Supporting SLA Provisioning in Grids by Risk Management Processes

von Diplom-Informatikerin

Kerstin Voß

aus Paderborn

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
Doktor der Ingenieurwissenschaften
- Dr.-Ing.-

genehmigte Dissertation

Promotionsausschuss:

Vorsitzende:	Frau Prof. Dr. A. Feldmann
Gutachter:	Herr Prof. Dr. O. Kao
	Herr Prof. Dr. J. Altmann

Tag der wissenschaftlichen Aussprache: 17. Juni 2008

Berlin 2008
D 83

Acknowledgements

When I look retrospectively at the last years, I am deeply grateful for all the support I received from my colleagues and family. I would like to use this opportunity to thank some special persons.

Special thanks goes to my doctoral advisor Odej Kao that he endorsed me all the time. He was always present to support and encourage me and available for discussion or advice. Working under his lead was really a great opportunity to research in a friendly atmosphere. Furthermore, I am very thankful that Jon Altmann agreed to review this thesis and gave helpful suggestions in the final phase.

I would like also to thank Bernard Bauer from the Paderborn Center for Parallel Computing who always has a sympathetic ear and relieved me from the financial reporting in EC-funded projects. The AssessGrid project has been funded by the European Commission. Thanks to all European tax payers for giving me the opportunity to research in this field. I am also indebted to Dominic Battré, Axel Keller, and Elmar Weber for their work and support in AssessGrid. Furthermore, I am deeply grateful to Iain Gourlay who was always willing to review and improve my written English.

Last but not least I would like to thank Jonas and my entire family. They have been backing me all the way and greatly supported me in hard times. I am happy that I found understandings from Jonas for all the time I was unavailable because of my research and work. Concluding, I can only thank him for the pleasure and diversion he brings in my life.

Abstract

Grid technologies have reached a high level of development, however core shortcomings have been identified relating to security, trust, and dependability of the Grid which reduce its appeal to potential commercial adopters. Users require a job execution with a desired priority and quality. In order to stipulate such requirements, *Service Level Agreements* (SLA) can be negotiated. These are a powerful instrument enabling the specification of the business relationships between service providers and service users in detail. However, providers are aware of various threats for SLA violations and are reluctant to adopt a mechanism which requires them to meet strict requirements and to guarantee associated quality constraints. If strict guarantees cannot be agreed by contract, many users prefer to operate their own resources instead of using the Grid. This is more expensive but they control their applications, which removes the issues of trust and ensures dependability concerning its successful completion. To establish a commercial Grid environment, it is essential that Grid providers are prepared to accept an approach involving SLAs with associated guarantees.

In order to enable providers to accept such SLAs, they need estimates of the likelihood that they are unable to fulfill an SLA, i. e. Risk Assessment. Furthermore the resource management should take into account such estimations when allocating resources or initiating fault-tolerance mechanisms, i. e. Risk Management. This work integrates risk awareness in the provider's processes which are involved in SLA provisioning:
During SLA negotiation they evaluate which resources can be used for service provisioning and estimate the *Probability of Failure* (PoF) of resources and of fulfilling the SLA. If the estimated PoF is too high, then, by applying risk reduction mechanisms, the provider may be able to reduce it sufficiently to accept the SLA. The estimated PoF will also be considered by the service provider and service consumer when determining the revenue and the contractual penalty. Compared to a service request requiring a relatively low quality of service, providing a more reliable service requires to receive a higher price since more guarantees have to be ensured. If a more reliable service is provided, the consumer might also define a higher contractual penalty. Thus, the PoF is an additional decision making element in the SLA negotiation since it enables end-users to compare different SLA offers by an objective measurement.
When providers have accepted an SLA, they have to be able to compensate for resource failures in order to prevent SLA violations. The usage of fault-tolerance mechanisms combined with risk awareness support Grid providers in this task. The Risk Management processes are interlaced with the resource management and thereby transparent for Grid service consumers. An important aspect of the Risk Management developed for the Grid are self-organising mechanisms, which initiate a fault-tolerance action or a chain of them, in order to manage resource instabilities or resource outages. Decisions are made on the basis of financial considerations, such as the profit margin, the cost for performing fault-tolerance, and the expected profit when executing a job. Taking into account such financial factors is of high importance for commercial Grid providers.

In conclusion, the integration of Risk Management in the processes of Grid providers is the initial step towards a risk aware Grid. It will increase transparency, reliability, and trust and provides an objective basis for decision processes in the resource management. Risk Management is integrated to address the SLA negotiation as well as the post-negotiation phase and thereby improves the SLA provisioning process in general.

Contents

1 Introduction **1**
- 1.1 Document Structure . . . 2
- 1.2 Grid Computing . . . 3
  - 1.2.1 Grid Definition and Categorisation . . . 3
  - 1.2.2 Computational Grids . . . 4
  - 1.2.3 Resource Management in Grids . . . 5

2 Problem Description **7**
- 2.1 SLA Support in Grids . . . 7
- 2.2 Applicability of Risk Management in Grid Computing . . . 10

3 Foundations **13**
- 3.1 Risk Assessment and Management . . . 13
  - 3.1.1 Definitions . . . 14
  - 3.1.2 Risk Management Standards . . . 16
  - 3.1.3 Risk Identification . . . 18
  - 3.1.4 Risk Assessment . . . 21
    - 3.1.4.1 Basic Measures of Risk . . . 24
    - 3.1.4.2 Specific Definitions of Risk in Different Fields of Application . 27
    - 3.1.4.3 Basic Types of Risk . . . 29
    - 3.1.4.4 Risk Assessment Methods . . . 30
- 3.2 Grid . . . 31
- 3.3 Service Level Agreements (SLAs) . . . 32
  - 3.3.1 Agreements for Service Usage . . . 33
  - 3.3.2 WS-Agreement – Agreement Structure . . . 35
  - 3.3.3 WS-Agreement – Context . . . 36
  - 3.3.4 WS-Agreement – Agreement Terms . . . 37
    - 3.3.4.1 Service Terms . . . 38
    - 3.3.4.2 Guarantee Terms . . . 38
  - 3.3.5 Agreement Negotiation . . . 40
- 3.4 Resource Management in Grids . . . 42

4 Requirements for Grid Service Providers **45**
- 4.1 Assumptions and Methodology . . . 45
  - 4.1.1 Assumptions and Dependencies . . . 45
  - 4.1.2 Models Used for Requirement Analysis . . . 47
- 4.2 Goal Model Based Requirement Analysis . . . 48
  - 4.2.1 Provider's Requirements during Negotiation . . . 52

4.2.2 Provider's Requirements in Post-Negotiation . . . 53
4.3 Functional Requirement Description . . . 54

5 Risk Management in the Grid 57
5.1 Applicability of Standard Risk Management Processes in the Grid . . . 58
5.2 Steps of a Grid Risk Management Process . . . 59
5.2.1 Defining Strategic Objectives . . . 60
5.2.2 Risk Identification . . . 61
5.2.3 Monitoring . . . 62
5.2.4 Risk Assessment . . . 62
5.2.4.1 Risk Analysis . . . 63
5.2.4.2 Risk Evaluation . . . 63
5.2.5 Risk Reporting . . . 64
5.2.6 Decision and Risk Treatment . . . 64
5.2.7 Residual Risk Reporting . . . 65
5.2.8 Summary . . . 65
5.3 Targeted Risk Management – Using Risk as Decisive Factor . . . 66
5.4 Provider's Risk Management . . . 68
5.4.1 Specification of Strategic Objectives . . . 69
5.4.2 Risk Identification . . . 70
5.4.3 Input Specification of Different Risk Factors . . . 71
5.4.4 Policies for Comparing Risks and Definition of Negligible Risks . . . 72
5.4.5 Notification of Grid Modules . . . 72
5.4.6 Decision Making and Risk Treatment . . . 74
5.4.7 Aggregation of Monitoring Information . . . 74
5.4.8 Risk Review . . . 76
5.5 Summary . . . 76

6 Risk Assessment and Underlying Model 79
6.1 Underlying Model of the Grid Fabric . . . 79
6.1.1 Grid and Grid Fabric . . . 79
6.1.2 Jobs . . . 81
6.1.3 Processing Jobs in the RMS . . . 84
6.1.4 Fault-tolerance Mechanisms . . . 85
6.2 Risk Assessment . . . 87
6.2.1 Introduction . . . 88
6.2.2 Dynamic Risk Assessment Model . . . 90
6.2.2.1 Computation of Equation 6.14 . . . 94
6.3 Procedure of Defining Risk Management Processes . . . 95

7 Risk Management During SLA Negotiation 97
7.1 Purpose of Integrating Risk Management During Negotiation . . . 98
7.2 Focusing on PoF in Resource Reserving . . . 100
7.2.1 Resource Pre-selection . . . 101
7.2.2 Risk Awareness and Risk Reduction . . . 103
7.2.3 Final Decision and Make Reservation . . . 107
7.2.4 Example Measurements for Risk-Focusing Reserving . . . 109

7.2.4.1 Select Execution Slot as Candidate for R . . . 110
7.2.4.2 Find the Best Starting Point . . . 110
7.2.4.3 Estimate the Schedule's Quality Change . . . 113
7.2.4.4 Example Computation . . . 114
7.3 Combining Risk Awareness with Arbitrary Scheduling Strategies . . . 116
7.4 Risk Reduction during Negotiations . . . 119
7.5 Risk Acceptance, Avoidance, and Transference . . . 125
7.6 Reservation Process in the Renegotiation Phase . . . 127
7.6.1 Modified Revenue or Penalty Fee . . . 127
7.6.2 Modified Probability of Failure . . . 129
7.6.3 Modified Service Terms . . . 129
7.6.4 Modified Guarantee Terms/Service Level Objectives . . . 130
7.6.5 Conclusion . . . 130
7.7 Recapitulation of Risk Management During SLA Negotiation . . . 130

8 Risk Management in Post-Negotiation Phase 133
8.1 Purpose of Integrating Risk Management Post-Negotiation . . . 134
8.1.1 Monitoring . . . 135
8.1.2 Initiation of Risk Management . . . 136
8.1.3 Benefiting from Risk Reduction Performed during Negotiation . . . 137
8.2 Risk Management Initiated by Monitored Instability . . . 139
8.2.1 Evaluation of Migration Effects . . . 139
8.2.2 Evaluation of Other FT-Mechanisms . . . 142
8.2.3 Decision Process . . . 144
8.2.4 Example . . . 144
8.3 Modification of Evaluation Formulas . . . 147
8.3.1 Simplification of Evaluation . . . 147
8.3.2 Recursive Checks . . . 148
8.3.2.1 Concept of Recursive Checks . . . 148
8.3.2.2 Termination Condition of Recursive Checks . . . 150
8.3.3 Example of Recursive Evaluation . . . 150
8.3.3.1 No Recursive Evaluation . . . 152
8.3.3.2 Recursive Evaluation . . . 152
8.3.4 Runtime Analysis . . . 154
8.3.4.1 Maximum Number of Recursion Levels . . . 154
8.3.4.2 Total Operating Cost . . . 156
8.4 Reacting After Resource Failures . . . 156
8.4.1 Using Planned FT-Mechanisms . . . 157
8.4.2 Generate New Schedule . . . 159
8.5 Recapitulation of Risk Management in Post-Negotiation Phase . . . 162

9 Evaluation Results 165
9.1 System Design . . . 165
9.1.1 AssessGrid . . . 165
9.1.1.1 Broker and End-user Layers . . . 167
9.1.2 SLA Negotiation . . . 168
9.1.3 Resource Management . . . 170

9.2 Evaluation of Risk Assessment . . . 171
9.3 Basic Scenario and Parameters . . . 172
9.3.1 Jobs . . . 173
9.3.2 Revenue and Penalty Fee . . . 175
9.3.3 FT-Mechanisms . . . 176
9.4 Evaluation of Applying Risk Management . . . 177
9.5 Other Scenarios and Parameters . . . 180
9.6 Contextualise Results with Risk Management Definitions . . . 181
9.6.1 Risk Management in General . . . 182
9.6.2 IT- Risk Management . . . 184

10 Related Work **187**
10.1 IT-Risk Management in Companies . . . 187
10.2 Analysing Stabilities of Resources . . . 189
10.2.1 Grid'5000 . . . 189
10.2.1.1 Difference to the Risk Assessment Model Developed . . . 192
10.2.2 Los Alamos National Laboratory (LANL) . . . 192
10.2.3 Failures of Hard Disks . . . 194
10.3 SLAs in Grids . . . 194
10.3.1 Overview . . . 195
10.3.2 Brokers . . . 196
10.3.3 Providers . . . 198
10.3.4 Resource Management Systems . . . 198
10.3.4.1 Provisioning of SLA/QoS . . . 198

11 Conclusion **201**

List of Figures **205**

List of Tables **207**

12 Bibliography **209**

Chapter 1

Introduction

It is well-known that Moore's law predicts that the computational speed increases exponentially. In particular in the sixties the speed even doubled every eighteen months [Scha 97]. The rate of increase has since slowed down, however, the exponential growth has evolved into a self-fulfilling prophecy [Moll 06]. However, despite the increases in computational speed, the requirements for solving challenging problems often still exceed the available computing power [Fost 00b]. Examples of such challenging problems are weather forecasts, medical image analysis [Baga 07, Mont 03], and can be found in the scope of modelling and simulation engineering of complex scientific processes. Users require high-end compute capabilities for such compute-intensive applications. These should be available for arbitrary users, independent of location or time. The first solution to offer access to heterogenous geographically distributed supercomputers was metacomputing [Smar 92]. The Grid concept, first presented by Ian T. Foster and Carl Kesselmann in 1998 [Fost 98], extends the metacomputing idea by widening the term of resources: the Grid combines arbitrary compute resources and not only supercomputing clusters and it further provides access to network connections, data repositories, sensors, or even human resources. In analogy to the electric power grid, individuals as well as (commercial) organisations should have access to transparent, pervasive, reliable, and also cheap computational power.

Most technical problems to support Grid computing, like the virtualisation of resources, have now been solved. Enterprises, which have identified the early market potentials of Grid computing, have been keen observers and active participants in the scientific research and development of Grid technologies and architectures. For example IBM [Witt 05], Hewlett Packard, or Fujitsu Siemens [Schn 05] became active in Grid research projects and standardisation organisations to influence developments and to benefit from experience of other Grid developers. Nowadays, the first steps towards Grid commercialisation can been seen since IBM and Hewlett Packard already offer commercial Grid services while Oracle has developed Grid enabled database solutions. The market of Grid computing is growing and shows a positive forecast for the next few years. Worldwide Grid spending is expected to grow from $1.8 billion in 2006 to approximately $24.5 billion in 2011. By 2011, Grid spending will account for 1.8 percent of total worldwide IT spending [Copo 06]. The expectations for the market are encouraging and potential end-users are already aware of the existence of Grid technologies. However, in order to encourage widespread commercial adoption of Grid technologies, significant obstacles have to be removed. The biggest obstacles are discussed in detail below.

Although using the Grid is often cheaper than maintaining one's own resources, end-users often are unwilling to give up control over their applications. In particular, the objection is based on

security issues in relation to the reliability of Grid services and industrial espionage [Biet 06]. In order to ensure resource provisioning with the desired security and *Quality of Service* (QoS), the current best-effort service is not sufficient. *Service Level Agreements* (SLAs) have been developed, which describe the user's individual requirements, and are negotiated between consumers and providers [Saha 03]. An SLA is a powerful instrument to describe all aspects of and obligations within a business relationship. The content of an SLA may be customer specific defined and each service aspect can be combined with a guarantee. The provider has to perform the service with the required quality, as defined by service guarantees, in order to fulfill the contract. If any guarantee is not met, the provider has to respond to this SLA violation by paying a penalty fee which may be defined for the complete SLA or for a single aspect of the service. The penalty fee is necessary both in order to protect customers from SLA offers made by unreliable providers and to compensate for them for losses incurred as a consequence of an SLA violation. However, agreeing to pay a penalty fee is a business risk for providers and consequently they may be reluctant to agree an SLA. Hence, providers negotiate SLAs with caution since a crash of multiple resources might lead to significant financial losses in the form of penalties. However, if providers only accept SLAs with low guarantees or a low penalty, customers are not willing to accept these, since either they do not get the required quality or in the case of an SLA violation their loss would not be covered by the penalty fee. This predicament can be solved by supporting the provider during the SLA negotiation with information about the probability that it has to pay the penalty for the SLA, i. e. the probability of an SLA violation. Such probability information is beneficial for providers not only during negotiations but also during system operation, since it offers new opportunities for the provider to prevent SLA violations or to find the most profitable solution. The process of assessing a probability and using it to inform decision support for initiating reactions is known as *Risk Management* . This thesis works on the integration of Risk Management processes in the Grid in order to establish the usability of SLAs and enable providers to accept SLAs which have adequate revenue and penalty in comparison to the provider's business risks.

1.1 Document Structure

This thesis is structured as follows. Chapter 2 presents a detailed problem description. In order to describe the problem in depth, understanding Grid concepts, especially from the provider's perspective, is essential. On account of this, the next section of this chapter introduces general Grid ideas. After the problem description the foundations for this thesis are described in Chapter 3.

The work for this thesis started with analysing the requirements for the Risk Management processes in the Grid. The results can be found in Chapter 4. Since no standard Risk Management process can be integrated in the workflows of Grid providers, Chapter 5 presents the Grid Risk Management process which was derived from the Risk Management standard of the *Federation of European Risk Management Associations* (FERMA) [FERMA 03]. As mentioned in the introduction, the Risk Management consists mainly of assessing the probability or risk and initiating activities according to the assessment. Chapter 6 describes the risk assessment methods and the underlying model of the Grid fabric and SLA provisioning. The Risk Management supporting SLA provisioning for Grid providers can be divided into two

phases: during SLA negotiation and in the post-negotiation phase. According to this differentiation the Risk Management activities and decision processes are subdivided into Chapter 7 and Chapter 8. Evaluation results are shown in Chapter 9.

Since the idea of using Risk Management processes in the Grid is completely new, comparing the thesis with related work in Chapter 10 focuses on related ideas in the scope of general IT-Risk Management, preventing SLA violations, and estimating resource reliability. The thesis ends with conclusions of the work in Chapter 11.

1.2 Grid Computing

In this section the ideas of Grid computing are described along general lines in order to form a basis for the detailed problem description in Chapter 2. Section 1.2.1 presents the general definition of the Grid as well as the categorisation of different Grid types: *computational* Grid, *data* Grid, and *service* Grid. Since this work focuses on the computational Grid, Section 1.2.2 provides a more detailed discussion on this type of Grid. Managing several resources and assigning Grid jobs to them is realised at the Grid provider site by a *Resource Management System* (RMS). Since support for SLAs has to be realised in an RMS, the main components of a RMS are presented in Section 1.2.3.

1.2.1 Grid Definition and Categorisation

The Grid is a widely used term for the integration of heterogeneous resources. Ian T. Foster and Carl Kesselman gave the following definition in 1998 [Fost 98]: *"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."*

After intensive work in this field, they redefined their definition of the Grid in cooperation with Steve Tuecke in 2000 [Fost 01] stating : *"The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organisation."*

According to these definitions, integrated Grid resources are as heterogenous as imaginable (from mobile devices to desktop computers, clusters, and software up to human resources). Grids are not usually categorised according to the types of Grid resources, rather the intention of the Grid utilisation is used to define several Grids categories so that the main Grid types are the *computational Grid, data Grid*, and *service Grid* [Krau 02].

In the computational Grid jobs primarily require computational power. Bundling thousands of powerful compute resources of clusters or working pools via the Internet results in a Grid of resources, which has a higher aggregated computational capacity than one single machine can

provide. Hence users can execute performance-challenging applications in a Grid environment instead of buying their own resources. Making resources available in a Grid reduces the wastage of unused computational power since geographically distributed users can execute jobs anywhere in a Grid. To support users by providing simple access to run applications on geographically distributed resources, Grid middleware solutions like Globus Toolkit [Glob 97, Fost 05b, Globus 08] or Unicore [Unicore 07] have been developed. Using such middleware, users do not have to concern themselves with the specific resources on which their application is executed. The resource selection and allocation is completely hidden resulting in a transparent Grid and significantly simplifying the utilisation. The middleware cooperates with RMS which are responsible for a set of resources on a Grid site, i. e. the resources which are available from one provider or organisation.

Data Grids provide controlled sharing and management of geographically distributed data repositories. Grids of this category conform to the idea of a data warehouse distributed in a *Wide Area Network* (WAN). Jobs in data Grids gain access to new information and often correlates data from different sources. Data Grids are often used by scientific users whose data-intensive applications need to transfer terabytes or petabytes of data between storage systems [Allc 00]. Furthermore, geographically distributed scientific or engineering applications and users in data Grids requires read access to gigabytes or terabytes of data in order to visualise or analyse them. To realise the data transfer and access, data Grids provide specialised data management features like replica management [Stoc 03, Carm 02]. Note that data-intensive applications, such as a data analysing job, might also be compute-intensive. As a result, data Grids have to provide additional features of a pure computational Grid, like resource discovery.

Service Grids ‘‘*. . . provide services that are not provided by any single machine.*” [Krau 02]. So applications using service Grids combines the utilisation of different resources. The usage of service Grids is differed between on demand computing, collaborative computing, and multimedia computing.

1.2.2 Computational Grids

Computational Grids are further classified according to the utilisation of the aggregated capacity. Grids for which it is important that the computational power is very high only in several time frames (and not permanently), belong to distributed supercomputing Grids. There supercomputers are connected in order to reduce compute times of applications. An example application for a distributed supercomputing Grid is computing weather forecasts. In contrast to a distributed supercomputing Grid is a high throughput computing Grid, which bundles arbitrary compute resources. It increases the completion rate for a row of jobs whereas the average, rather than, the peak performance is in focus. The most famous project for high throughput computing Grids is SETI@home [McCo 01, Seti 08], which *Searches for Extraterrestrial Intelligence (SETI)* since 1999. In this project arbitrary (personal) computers, which are connected with the Internet, can be integrated in the search by installing a program that automatically downloads and analyses radio telescope data.

The focus of this work is on computational distributed supercomputing Grids since they are the basis for other Grid types. In order to combine as much computational power as possible,

geographically distributed clusters are connected within a Grid infrastructure. To provide cost effective cluster solutions, the machines are often built up from homogeneous commodity-off-the-shelf components. Hence the Grid consists of sets of (hardware) homogenous cluster nodes and of heterogenous cluster systems. Even if the hardware of compute nodes of the same cluster do not differ, they must not to be completely homogenous since the configuration, installed software, software versions, and available services can widely differ. The access and usage of Grid resources is organised by resource management systems (RMSs) in the Grid fabric layer which interact with Grid middleware solutions.

Note that a high-throughput computational Grid can consist of several workstations for example widespread over a campus. In such Grids the heterogeneity of resources is much higher. Another disadvantage of campus-wide Grid sites is the connectivity of different workstations which is an important criterion when executing parallel jobs. The interconnection between nodes of a cluster is generally very fast in contrast to the connectivity within a Ethernet, so executing a communication-intensive parallel job on nodes of one cluster will be significant faster than on a campus-wide Grid site.

For this work the performance of nodes is not crucial. Integrating risk assessment and management in RMSs for dedicated clusters is the first step, since in a Grid site of pool-clients the local utilisation of the resource is a big uncertain factor. Accordingly, in this work the integration of Risk Management in a distributed supercomputing Grid is described. In order to use the developments presented in this thesis in Grids of workstations, the risk assessment methods just have to be enhanced to consider probabilities that resources are locally used and not available all the time for a Grid computation job.

1.2.3 Resource Management in Grids

The Grid layer, in which the Grid resources are located, is called the Grid *fabric* [Fost 01]. Groups of resources typically belong to autonomous administrative domains which are under the control of local Grid providers. Providers make the resources available for Grid utilisation and are responsible for managing job executions. The central component for handling distributed resources on one Grid site is the RMS (see Figure 1.1).

Its main tasks are to accept requests for resource utilisation, match these requests to suitable resources, and initiate the job execution on selected Grid resources. The scheduling process is quite complex since it has to consider the resource requirements of the Grid job as well as the overall suitability of resources for the job. The job-resource-mapping process results in a schedule which assigns jobs to resources.

In the Grid there are two contrasting approaches to the scheduling process: *queuing based* and *planning based*. A queuing based scheduler inserts job requests in a queue and according to a specific policy (*First Come First Serve* (FCFS), *Earliest Due Date* (EDD), etc.) [Panw 88] one job after another is dropped out of the queue and assigned to free resources (see Figure 1.2). This matching is executed each time a resource becomes available or a new job arrives and enough resources for its execution are free. Jobs using multiple resources are removed from the queue and allocated to resources when an adequate number of the appropriate types of resources are available. Planning based schedulers make the matching immediately after they have received the request so that the assigned time slot for the job execution is planned in

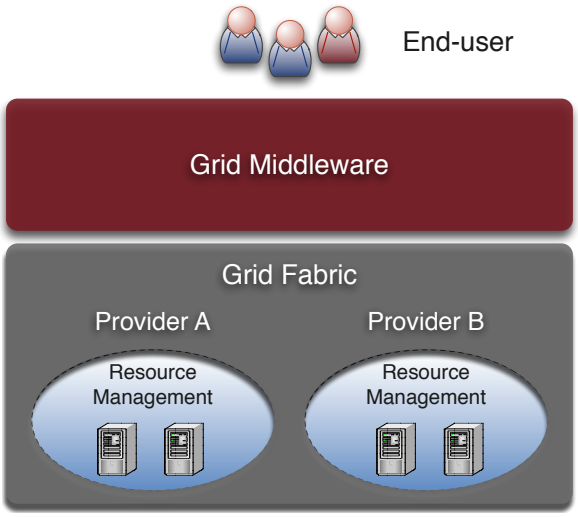


Figure 1.1: Coarse-Grained Grid Architecture – Overview of Grid Layers and Resource Management

advance. Such time slots, which are assigned for a job execution in the future, are referred to as *advance reservations*. A comparison of queuing and planning based systems can be found in [Hove 03].

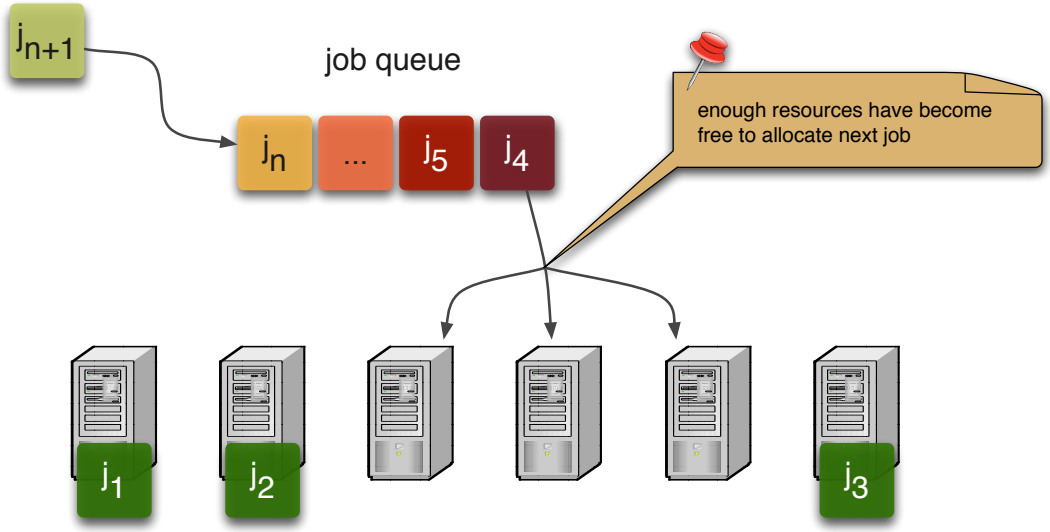


Figure 1.2: By and By Job Allocation in Queuing-Based Systems

Chapter 2

Problem Description

This chapter details the problem of establishing *Service Level Agreements* (SLAs) in Grids. Section 2.1 illuminates the gap between SLA as a concept and as an accepted tool from the perspective of service consumers and service providers. The work in this thesis closes this gap by integrating Risk Management activities into Grid processes. Section 2.2 discusses the applicability of Risk Management in Grids to establish the SLA usage.

2.1 SLA Support in Grids

A desire to drive Grid technology from academic research to a viable technology platform for a wide range of users has resulted in considerable focus on *Quality of Service* (QoS) issues in the Grid. Grid users expect that their jobs and data will be processed in the desired manner and on time. In the computational Grid, jobs running a simulation or parallel application often require a specific number of compute resources at the same time. Consequently, the job execution can only be successful if a specific running environment is provided. The job's individual requirements for its execution environment address not only the number or configuration of resources; in particular, usually the Grid end-user needs the results by a specific time. Hence, they can only accept a fixed time frame for the execution, i. e. defined through earliest possible start time and latest admissible completion time (see Figure 2.1).

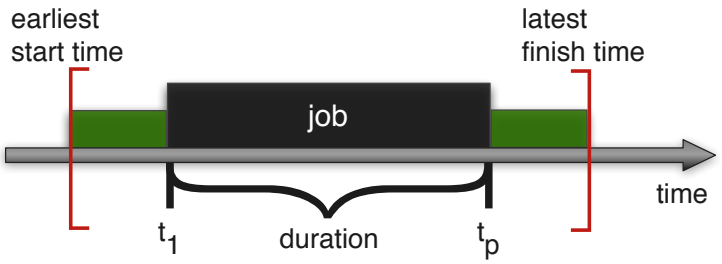


Figure 2.1: Job Slot Bounded by Earliest Start Time and Latest Completion Time

To contractually determine QoS for a job execution between customers and providers, SLAs are a powerful instrument [Saha 03]. Agreeing SLAs with providers is beneficial for a wide range of Grid end-users working in a scientific institute or a commercial environment. For commercial Grid end-users SLAs are not a luxury, they are an essential means of specifying

fixed guarantees and ensuring they receive penalty payments as compensation if the agreed QoS guarantees are not provided. Considerable progress has been made in the area of SLA specification, but commercial Grid utilisation has not been really established yet. The main reasons from the users' and the providers' perspective are described below.

End-users are reluctant to agree SLAs since they are aware that a hundred percent success guarantee can not be given. Service guarantees are of crucial importance to commercial end-users and any losses that may result from an SLA violation need to be accounted for. For example, an SLA violation could result for service users in a delay in software development, financial losses or damage to reputation. Due to these potential losses end-users may either be unwilling to use the Grid or may demand high penalties in the event of SLA violation, in order to reduce their losses. Figure 2.2 depicts the situation of end-users.

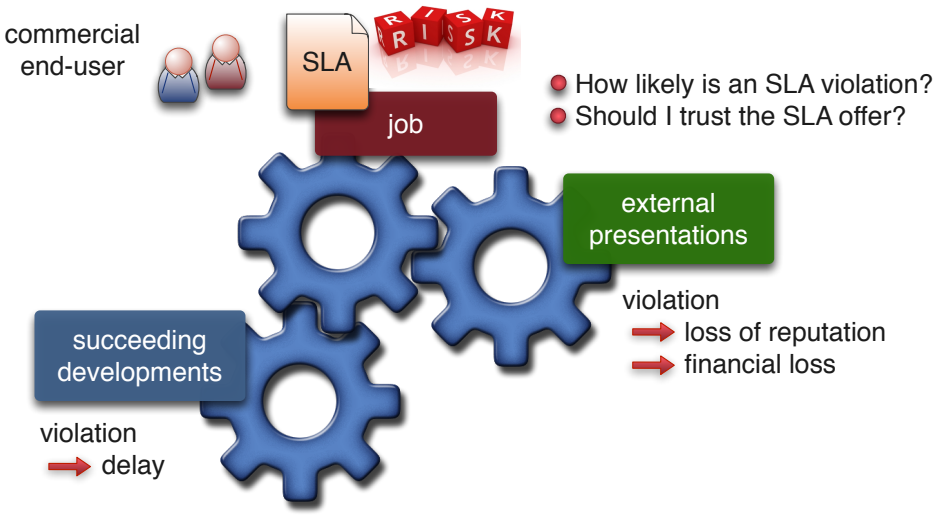


Figure 2.2: Problem of Customer Perspective – Will SLA be Fulfilled?

Providers are cautious about adopting a system on agreeing SLAs since penalty payments can result from a variety of failures and un-availabilities: compute nodes can fail, networks can break down, experts may not be not available, or resources can be overbooked. Being aware of such possible events, providers may be unwilling to agree an SLA since they can not estimate the circumstances of the future job execution in detail (see Figure 2.3). Note that a provider can accept without risk guarantees to provide specific resources in an *arbitrary* time frame, i. e. without a deadline, if it operates suitable resources. However, time constraints are often the most important guarantees in SLAs since customers cannot wait arbitrarily long to receive their results. Consequently, the provider has to answer the following question: *Will it be possible to make the requested resources available for the specified job duration in order to ensure the deadline is met?* Hence, it has to determine in advance whether enough resources – fulfilling the resource constraints of the SLA – can be assigned to the job to ensure completion before the deadline.

The scheduler of the *Resource Management System* (RMS) is the component responsible for generating the mapping of jobs to resources. Here, it is important to differentiate between

Figure 2.3: Problem of Provider Perspective – Is it Possible to Fulfill the SLA?

the queuing and planning based approaches (see Section 1.2.3). A queuing based scheduler performs the mapping each time a resource is freed by a completed job. Consequently, the provider has to estimate when resources will become free and when the job considered will be removed from the queue. Using a planning based scheduler, such predications are not necessary since an advance reservation is made for the job during the SLA negotiation. Since through making an advance reservation suitable resources are reserved a priori, the certitude for the provider increases that the job can be completed before the deadline. For QoS provisioning in Grids, advance reservations should be supported by the RMS [Al A 04]. Hence in this thesis the assumption is made that a planning based system is used. The biggest uncertainty for fulfilling an SLA is the occurrence of resource outages or failures which do not depend on the usage of a queuing or planning based system. For example in the Grid'5000 nodes have only an average uptime of $\approx$ 45 hours, followed by an average downtime from $\approx$ 14 hours [Iosu 07]. In contemporary systems the threat of resource outages is not estimated during SLA negotiations, although it is an important decision criterion for providers.

A means for handling such resource failures is the initiation of *Fault-Tolerance* (FT) mechanisms like checkpointing [Ande 81] and migration [Hein 05, Wrze 05]. Furthermore an SLA violation can be prevented by executing the job redundantly on different resources. Redundant job execution is expensive since twice the required resources are consumed. In addition redundant job execution is no guarantee that the SLA is not violated in the case of resource failures. If the job is redundantly executed on nodes of the same cluster rack and the power supply for this rack drops out, the SLA is also violated. Similar problems exist in the migration to alternative resources since these might also fail because of either a correlated or uncorrelated error. In this case the remaining time might then be insufficient to perform a further migration. These examples show that by using FT-mechanisms an uncertainty still exists as to whether the SLA can be fulfilled. This uncertainty is reflected in failure rates of jobs in Grids which are quite high: for example in the Grid3 still 27% jobs fail when per-

forming 5-10 retries [Dumi 05]. Hence the stability of resources should be considered prior to resource allocation, while initiation of FT-mechanisms is an additional precaution.

SLAs are a powerful instrument to define QoS attributes for job execution in the Grid. In particular, the community agreed on this approach to define contracts for service usages, as the developments of the *Open Grid Forum* (OGF) [OGF 08] indicate – the WS-Agreement [Andr 07] is in wide-spread use. However, Grid end-users are likely to be reluctant to agree SLAs since they cannot estimate the likelihood of an SLA violation. The lack of such information means that end-users are not able to estimate the effects of choosing to execute a job with a particular provider – and even of using the Grid in general. Grid providers only are also cautious about adaption of an SLA-based approach because they are aware of failures which could result in an SLA violation. Since they cannot estimate the business risk of agreeing an SLA, they are only willing to accept a low level of guarantees in relation to the penalty which is often not acceptable for commercial end-users. In conclusion, the usage of SLAs is not established and it is not an accepted tool in commercial Grid service provisioning.

2.2 Applicability of Risk Management in Grid Computing

Risk Management is important in many disciplines such as statistics, economics, biology, engineering, psychology, systems analysis, or operations research [Carnegie 05]. H. Felix Kloman, a Risk Management expert, has described the meaning of Risk Management as follows: To social analysts and politicians it is the management of technology-generated and environmental macro-risks that appear to threaten our existence. To bankers it is the sophisticated use of techniques as currency hedging and interest rate swaps. To insurance buyers and sellers it is the coordination of insurable risks and the reduction of insurance costs. To safety professionals it is reducing accidents and injuries. ‘‘*Risk Management is a discipline for living with the possibility that future events may cause adverse effects.*” [Carnegie 05] Traditionally, risk was seen as a *negative* force. Modern Risk Management treats risk as a *positive* force since opportunities are created where others shy away from possible dangers. Accordingly, risk offers both opportunity and danger. This interesting duality implies that Risk Management should not only be oriented towards *risk avoidance* rather it should take into account the potential benefits of *taking* certain risks.

At present, Risk Management techniques are not integrated in Grid environments. However, the gap between SLA as a concept and as an accepted tool in the Grid, which has been pointed out in Section 2.1, can be closed by integrating Risk Management across all Grid layers.

Grid resource providers are aware that their resources can fail in which case they have to compensate for the customer through a penalty. However they do not have mechanisms to evaluate the risk of agreeing an SLA for a specific job. Since the inherent risk of an SLA violation cannot be estimated in contemporary systems, resource providers do not offer SLA provisioning at the moment. Risk Management enables providers to make decisions based on facts: estimating the *Probability of Failure* (PoF) of an SLA and using it during negotiation as well as post-negotiation. PoF information will support providers in making the decision as to whether agree or reject an SLA request and further offers opportunities to initiate precautionary fault-tolerance in order to prevent SLA violations. If no spare resources are

available in order to compensate for all resource failures, the Risk Management supports the provider in finding in expectation the most profitable solution.

The estimated PoF can be published as an additional SLA information in order to gain a user's trust. By introducing risk parameters in negotiations, customers will have an increased level of trust in offers since an estimated PoF value is more trustable. In order to prevent that providers are lying, a reputation center on the broker layer can observe failure rates and PoFs published. It is then able to provide end-users an independent opinion of an accurate PoF. The PoF enters the Grid as a negotiable parameter. The availability of PoF information enabes the enforcement of Risk Management policies on all architecture layers. Integrating Risk Management and assessment techniques into the Grid fabric forms the basis for risk aware Grid services in all layers. Accordingly, the development and integration of Risk Management methods into the Grid fabric layer is the driving force of this thesis.

End-users are aware that SLAs cannot be guaranteed against failure because system failures can always occur. If resource providers or brokers offer their estimate of the PoF of an SLA, customers are primarily informed and are able to trade off between the benefits and the risk of accepting a particular offer from a specific provider or using the Grid at all, i. e. any Grid resource provider. Through this new risk aware approach customers are enabled to select the desired level of realistic guarantees under consideration of the cost they are willing to pay and the risk they are willing to accept. By integrating the PoF in SLA negotiations, new market mechanisms and policies can arise within the Grid, e. g. the calculation of a PoF-dependent revenue and penalty fee. Furthermore, the provider can define risk-specific policies regarding SLA negotiations and resource management.

In conclusion, offering the PoF for an SLA should not be seen negatively. The benefits are obvious: publishing the PoF in SLAs is more trustable for customers and new possibilities for revenue/penalty modelling are opened. Furthermore, trustable providers can prove their reliability and raise their reputation by good performance. SLAs can thereby become a really useable tool for providers as well as customers.

Note, that in business and industry, traditional risk assessment and management is seen from different point of views. Risk assessment and management address the work and performance of persons. The manager will see the resulting increase in efficiency and in revenue. People, whose work is evaluated and portfolio managed, are not enthusiastic since they see primarily the negative aspects of using Risk Management (for more details see [Koll 99, p. 14]). However, in the Grid the work and performance of machines is estimated which will not complain of using risk assessment and management.

Chapter 3

Foundations

This chapter present the foundations of the work. For the integration of Risk Management processes in the Grid, the basic concepts in this field are described in Section 3.1. Since the new field of application of Risk Management is Grid computing and the Grid fabric in particular, Section 3.2 and Section 3.4 complete the brief overview of the Grid and *Resource Management Systems* (RMSs) given in the introduction. *Service Level Agreements* (SLAs) are detailed in Section 3.3.

3.1 Risk Assessment and Management

Risk is a concept that relates to human expectations about the occurrence of future events. It denotes a potentially negative impact on an asset by depreciating some of its characteristic value. Risk arises and is triggered by some present process or by a future event. Generally, risk is understood to represent the *probability* of a loss implied by the threatening event. In professional risk assessment, the notion of risk *combines* the probability of events with the impact of those events. Thus, mathematically risk is equal to the *weighted average* of the expected losses with respect to their chances of occurring.

Risk Management is the process whereby organisations methodically address the risks attaching to their activities with the goal of achieving sustained benefit within each activity and across the portfolio of all activities. The focus of good Risk Management is the identification and treatment of these risks. The notion of risk has been traditionally understood as a *negative* force (also called *downside potential*) which should be avoided using Risk Management. Modern Risk Management treats risk as a positive force (*upside potential*): opportunities are created where others shy away from possible danger. Consequently, new Risk Management methods were required in order to take into account the potential benefits of taking certain risks.

In the following definitions of terms are given in order to consider risk assessment and management standards. These terms will be used in the whole work. Afterwards in Section 3.1.2 the process of risk assessment and management is described by comparing the Risk Management standard from the *Federation of European Risk Management Associations* (FERMA) [FERMA 03] and the *Australia and New Zealand* (AS/NZS) [ASNZS 99] standard, which are both established. Beyond the decision strategies, in Risk Management processes the risk identification and risk assessment form two important steps. Section 3.1.3 gives a

summary about approaches and methods which can be used in the risk identification phase. Section 3.1.4 details the foundations in scope of risk assessment.

3.1.1 Definitions

As the term indicates, Risk Management is based on risk values. The correct definition according to *International Organization for Standardization* (ISO) is:

Definition 3.1.1 (Risk [ISO 02])
Risk can be defined as the combination of the probability of an event and its consequences.

A fundamental issue in the characterisation and representation of risks is to properly and appropriately carry out the following steps:

1. Analyse the triggering events of the risk, and by breaking down those events formulate their adequately accurate structure.
2. Estimate the losses associated with each event in case of its realisation.
3. Forecast the probabilities or the possibilities of the events by using either
   - statistical methods with probabilistic assessments or
   - subjective judgements with approximate reasoning.

After the possible risks has been identified, it is necessary to assess them in terms of their

- potential severity of loss and
- probability or possibility of occurrence.

This process is called *Risk Assessment* (RA). The input quantities for risk assessment can range from simple to measurable (when estimating the value of a lost asset or contracted penalty associated with non-delivery) to impossible to know for certain (when the probability of a very unlikely event has to be quantified).

Definition 3.1.2 (Risk Categorisation [FERMA 03])
In general risks are driven by externally and internally driven factors. Further risks can be categorised into strategic, financial, operational, hazard, etc.

Additionally, risks can be divided according to their frequencies and appearances rates. *High-frequency events* such as hardware or software failures, performance bottlenecks or access violation can be easily measured. On the contrary, the probabilities and necessary information for a general frequency estimation for *low-frequency events* are hardly assessable. Example for such events are fire, natural disasters (flood, earthquake, . . .), or gross carelessness [OMah 05]. Consequently, the risk assessment model focuses on high-frequency events. The risk resulting from low-frequency events can be considered by an additional fixed number.

Definition 3.1.3 (Risk Assessment [ASNZS 99])
In scope of risk assessment the following terms are used to identify the sub-steps.

- ***Risk identification***: the process of determining in the system critical factors: what can happen, why, and how.

- ***Risk assessment***: the overall process of risk analysis and risk evaluation.
- ***Risk analysis***: a systematic use of available information to determine how often specified events may occur and the magnitude of their consequences.
- ***Risk evaluation***: the process used to determine Risk Management priorities by comparing the level of risk against predetermined standards, target risk levels, or other criteria.

Risk Management is the process of measuring or assessing risk and on the basis of the results developing strategies to manage that risk and control its implications. Managing a type of risk includes the issues of

1. determining whether an action – or a set of actions – is required, and if so then
2. finding the optimal strategy of actions to deal with the risk.

After the risk value is assessed, actions are taken to react on this information or prevent some consequences.

Definition 3.1.4 (Managing Risk [ASNZS 99])
In the scope of managing assessed risk values, following terms are standardised:

- ***Risk Management***: the culture, processes, and structures that are directed towards the effective management of potential opportunities and adverse effects.
- ***Risk Management process***: the systematic application of management policies, procedures, and practices to the tasks of establishing the context, identifying, analysing, evaluating, treating, monitoring, and communicating risk.
- ***Risk treatment***: selection and implementation of appropriate options for dealing with risk.
- ***Risk reduction***: a selective application of appropriate techniques and management principles to reduce either likelihood of an occurrence, its consequences, or both.
- ***Residual risk***: the remaining level of risk after risk treatment measures have been taken.
- ***Risk acceptance***: an informed decision to accept the consequences and the likelihood of a particular risk.

Note that the risk treatment includes transferring the risk to another party or avoiding the risk [Majl 06, p.14].

The risk calculation is based on several random variables. If one variable influences other variables, it *controls* the others [Koll 99]. Such controls are important to model real-world processes. "*Foisting a risk assessment process or model upon an organisation will not only change how opportunities or liabilities are assessed, but will significantly alter the way an organisation makes critical decisions. . . . , such decision-making processes typically lead to selection of 'winners' and 'losers'.*" [Koll 99, p. 12] Further it is not sufficient to only have an arbitrary risk assessment process. The quality of a risk assessment is important since a hastily constructed and poorly implemented 'Risk Management' do more harm than good. Accordingly, big efforts in a detailed analysis of the risk factors and a precise development will be in charge of the success of the risk assessment and management.

3.1.2 Risk Management Standards

One of the standards for Risk Management processes was published by the council of standards Australia and council of standards New Zealand Risk, AS/NZS 4360:1995. This was followed by the Canadian standard CAN/CSA-Q850-97 and was updated into the version AS/NZS 4360:1999 [ASNZS 99]. In 2001, Japan launched a guideline for developing Risk Management systems, called Japan Issued Standard JIS Q 2001:2001 [JIS Q 01]. It offers two advances: the formal definition of a Risk Management system and the introduction of continuous improvement. In 2002, the three UK Institutes AIRMIC[1], ALARM[2], and IRM[3] introduced their Risk Management standard which was published in 2003 from the *Federation of European Risk Management Associations* (FERMA) [FERMA 03]. The importance of the FERMA standard is shown by the fact that ISO released vocabulary guidelines for use in standards according to the FERMA definition [ISO 02].

The definitions of risk assessment and management procedures differ in several standards. The FERMA standard, depicted in Figure 3.1, has a more detailed concept than the AS/NZS, shown in Figure 3.2. However, they have many equalities. Both concepts are described in the following in order to form the basis for the risk management approach of this work.

The first step in both models establish the context of the Risk Management. The organisation's strategic objectives have to be considered in order to define the risk assessment and treatment. So criteria for the risk assessment will be established in the first phase. These criteria have to be selected carefully since they implicate the goals of the risk assessment. In business and industry the defined goals should not be too presumptuous in order to not discouraging responsible staff, but a too low level is also not advantageous because otherwise the necessity for the risk assessment and management is not clearly given. For business plans and industry this observation is comprehensible. However, in the Grid risk assessment the work of machines and not of people is estimated.

According to the defined context, in which the Risk Management should be active, a risk identification builds dependencies of possible problems, their impact, and their origin. This is essential to estimate several risk factors. After identifying and categorising problems with their origins and their consequences, the risk has to be computed for several problems. Therein the probability or likelihood for an event as well as its consequence are considered in the context of the defined risk model. In the AS/NZS standard this proceeding is named risk analysis (see Figure 3.2) including a task named *'estimate level of risk'*. In [FERMA 03] the process of the whole risk assessment contains also the identification process (see Figure 3.1). However, the risk estimation has the same tasks as the risk analysis of the AS/NZS Standard. The ordering of these sub-task is insignificant since in both models the risk assessment and identification process is cyclic.

An important step in both models is the risk evaluation which follows the risk estimation/analysis. The estimated levels of risks are compared against pre-established criteria. Therewith it is possible to filter negligible risks if their danger is negligible and their treatment may not be required.

[1]The Association of Insurance and Risk Managers, see http://www.airmic.com/
[2]The Association of Local Authority Risk Managers, see http://www.alarm-uk.com/
[3]The Institute or Risk Management, see http://www.theirm.org/

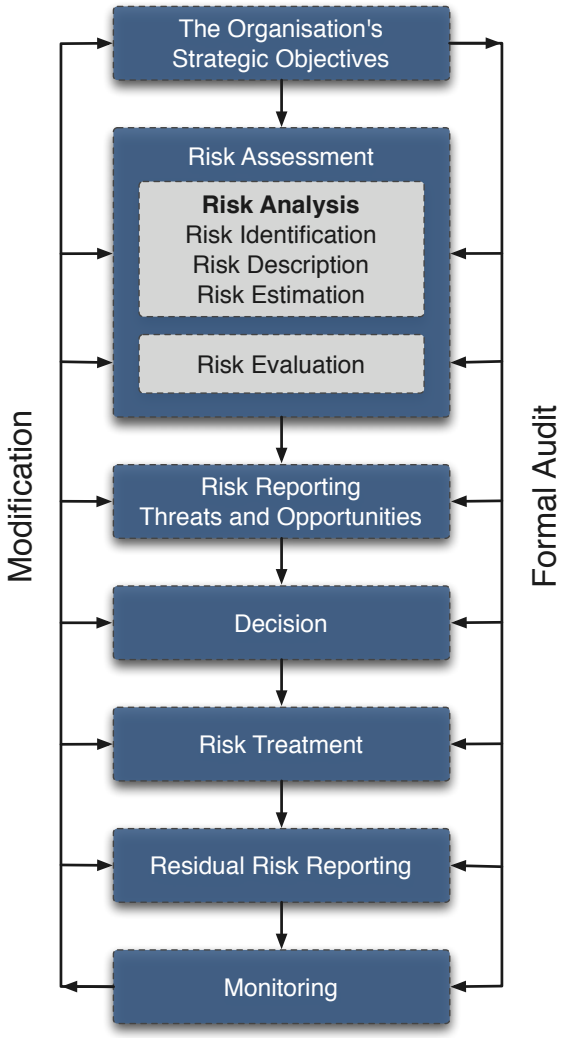


Figure 3.1: The Risk Management Process in FERMA [FERMA 03]

According to the AS/NZS standard the risk treatment is not defined clearly. It is mentioned that for middle- and high-priority risks a specific management plan has to be developed and implemented. The FERMA standard divides in contrast to AS/NZS the risk treatment and management precisely into several steps: After the risk assessment the risk is reported with its opportunities. On this basis the management modules make their decision whether and how to treat the assessed risk. Since often the risk can only be reduced, the residual risk has to be reported. If the residual risk is still too high, further risk treatment can be necessary. In the last step of the Risk Management process the system is monitored in order to identify new appeared risks.

Monitoring and reviewing the results of the risk assessment is mandatory for an dynamic and adaptive procedure. Additionally, the feedback from internal and external stakeholders realises a Risk Management conforming to organisation's objectives.

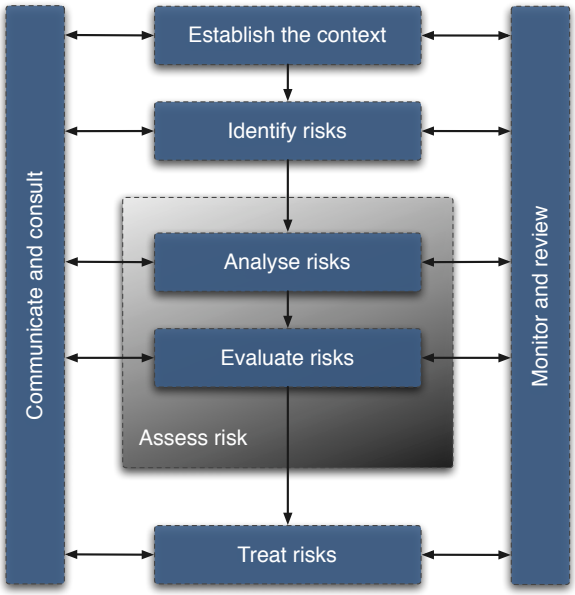


Figure 3.2: The Risk Management Process in AS/NZS [ASNZS 99]

3.1.3 Risk Identification

As described in Section 3.1.2 the first step of a Risk Management process is to establish the context or clarifying the objectives of the Risk Management in order to define the basis on which the risk is evaluated. Afterwards the risk identification follows which processes the results of the context definition. In general, risk identification is an essential chain of the Risk Management framework: it is crucial to identify the relevant risks before assessing them. Therefore the risk identification is an important sub-process in the Risk Management, since an adequate addressing and presenting of the risk-related issues in the risk identification process forms a basis for generating convenient and reliable risk calculations and interpretations.

The common risk identification methods are presented in the following [Majl 06].

Objective-Based Risk Identification
Organisations and project teams have well defined objectives. They define risk as an event that may endanger achieving one of their objectives partly or completely. This type of risk identification method is employed by the *Committee of Sponsoring Organisations of the Treadway Commission* (COSO) which builds the basis of the Enterprise Management Integrated Framework [COSO 04]. This framework defines all tools (principles, concepts, as well as a common language) for identifying, assessing, and managing risk.

Scenario-Based Risk Identification
This approach incorporates scenario analysis in which different scenarios are created to represent the alternative ways through which an objective can be achieved, and also to analyse the interaction of the forces in the considered environment (e. g. in a market or battle). In this setup, any event that triggers an undesired scenario is considered as a risk.

Taxonomy-Based Risk Identification
This method presents a *breakdown* of possible risk sources according to certain criteria and their degrees of importance. Based on the taxonomy and knowledge of best practices, a questionnaire is developed and its results are compiled. Following this technique, the taxonomy methodology is based on the assumption that the answers to the questions can reveal the potential risks. Thus, creating a well-targeted questionnaire, whose results can readily be digested and turned into useful information and knowledge, is an important objective. This type of risk identification is effectively used in the software industry [Carr 93].

Common-Risk Checking
In several industries, there are lists available with known risks which might be identified because of historical events or experts' knowledge. Each item in the list represents a threat which is checked whether it applies to the particular situation. In this case, in the risk identification phase consists only of evaluating which listed risks have to be considered. For example, the Common Vulnerability and Exposures list[4] presents known risks in the software industry.

In practice the risk identification phase is started with analysing the sources of the problems or with characterising the problems themselves. In the case that either the source or the problem is known, their relationship can be analysed by formulating 'how' the source could trigger the unwanted event.

Source Analysis:
Sources to be identified can be either *internal* or *external* to the system which is under consideration of the Risk Management process. While internal risks can be managed by applying an appropriate control in the system, risks originating in external sources may not even be addressable with respect to the performance of the system. Examples of risk sources are stakeholders of an R&D project (external for the researchers involved, internal for the operating management), employees of a brokerage company (internal), or the weather over an airport (external).

Problem Analysis:
All the possible problems and hazards that might occur in the system under examination have to be considered, of which the triggering event can be kept under control. In this scope risks are related to identified threats. Threats can also exist with various entities, most importantly with stakeholders, customers, and legislative bodies such as the government.

Examples of threats include not being able to complete a computational task, the threat of errors in the results due to systems failure, the threat of late completion of tasks (making the results obsolete), the threat of break-downs in communication networks, etc.

It follows an example for clarifying the differences of risk, threat, and risk source.

Example 3.1.1
If stakeholders (risk source) start withdrawing during a project, then this can endanger the future funding of that project (risk), which can eventually make it a failure and a financial loss

[4] available under http://cve.mitre.org/

(threat). On the other hand, there is a possibility that confidential information is stolen (risk) and misused (threat) by employees (risk source), even if the Grid operates with high security, and uses a closed network. Finally, the Grid computing service centre experiencing a heavy electrical storm (risk source) is eager to know whether lightning will strike (risk) the network, and if so then whether it will cause a crash of the system (threat) or only a minor/shorter interruption of services.

The methodology chosen for identifying risks primarily depends on corporate culture, industry practice and standards, as well as the requirement of compliance. The identification methods are formed by templates or rules, which make them ready to computable. The important aspect in the risk identification are not the templates or rules themselves, rather the methodology applied to develop them and their functioning in the identification of the risk source and of the problem or threat.

If either the source of the problem are known, or the events that the source may trigger and which then lead to the problem, then the risk identification phase can be performed by an appropriate analysis. It is obvious that a proper formulation of the causes of the risks play a crucial role. However, doing either source analysis or problem analysis, it has to be kept in mind that describing the actual system of relationships in a realistic model can be very complex. To manage these types of difficulties, the results can be employed of a recently developed field, called *Morphological Analysis* (MA) [Zwic 98].

MA was developed by Fritz Zwicky, a Swiss-American astrophysicist and aerospace scientist while working at the *California Institute of Technology* (CalTech), as a method for structuring, representing, and analysing the total set of relationships associated with multidimensional, non-quantifiable, complex problems [Zwic 67, Zwic 69]. Originally, Zwicky applied this method to a wide range of fields which include the classification of astrophysical objects, the development of jet and rocket propulsion systems, and the legal aspects of space travel and colonisation.

Definition 3.1.5
Morphological Analysis (MA) is a method for rigorously structuring and analysing the total set of relationships of a system that are embedded in inherently non-quantifiable sociotechnical problem complexes – also called 'wicked problems'. Morphological analysis is carried out by

1. developing a discrete parameter space of the problem that should be formulated and
2. defining relationships between the parameters on the basis of internal consistency.

Such an internally linked and thus structured parameter space is called a *morphological field*. With proper computer support a morphological field can be treated as an inference model which can provide an efficient decision support tool for risk identification.

When performing risk assessment precisely, problems of representing and manipulating complex structures inevitably appear. Based on these structures it is necessary to develop policy fields and future scenarios, which further complicates the risk assessment task by presenting a number of difficult methodological issues. The question that needs to be addressed in this environment and that MA seeks to answer is the following:

How can developers of Risk Management processes represent and put judgmental processes on a sound methodological basis?

MA essentially imitates the evolution of scientific knowledge through human history. Scientific knowledge has been developed through recurrent cycles of *analysis* and *synthesis*: Every *synthesis* is built upon the results of a preceding analysis and theoretical development, and every *analysis* requires a subsequent synthesis in order to verify its results and correct its deficiencies. In the following, the original approach of morphological analysis formulated by Zwicky in 1969 is briefly presented. The process of MA consists of the following five iterative steps [Zwic 69]:

1. The problem has to be carefully and concisely formulated.
2. All parameters of the model have to be determined that might be of importance for the solution. Those parameters have to be localised and analysed with respect to their range and sensitivity.
3. Using the parameters as attributes, the so-called *morphological box* is constructed of the problem. This is a multi-dimensional matrix that is defined by the a priori defined attributes. In essence, the morphological box contains all the potential solutions of the problem, where each solution uniquely defines the values of all the parameters.
4. Considering all feasible solutions that can be represented by the morphological box, these are closely scrutinised and evaluated with respect to the goals that is aimed to achieve. Obviously, this is the phase when Risk Management is applied and the risk assessment attitude is incorporated. That is, based on the goals, a mixed strategy with following elements is defined
   - risk avoidance,
   - risk transference,
   - risk mitigation, and
   - risk acceptance.
5. Comparing the feasible solutions, the optimal – or, in lots of cases the most suitable yet attainable – solutions are selected. Provided that the necessary resources are available and the constraints of the model are satisfied, those solutions are practically applied as well. Usually, the last step, where the theoretical solutions are tested and applied in practice, requires a supplemental morphological study.

3.1.4 Risk Assessment

This section presents the key definitions of the conceptual framework to provide a consistent development of the risk assessment processes in the Grid. Of particular importance in this part is the introduction of a measure of risk.

In the professional Risk Management risk is based on probabilistic considerations and describes not only the probability of a bad event, rather it combines two measures, the *loss* that can happen, and its *probability*. As opposed to this approach, there exist other representations of risk which are applied in other areas. In particular, mathematically, the risk of an event

is identified as the probability of its happening. In this case, risk only presents one measure (probability). Details are given in Section 3.1.4.1.

This section is based on a deliverable developed during the AssessGrid project [Majl 06, chapter 6].

After possible risks have been identified during the risk identification phase, they have to be assessed in terms of their

- potential severity of loss and their
- probability of occurrence.

These quantities can either be simple to measure (e. g. in the case of the value of a lost asset or contracted penalty associated with non-delivery), or impossible to know for sure (when the probability of the occurrence of an unlikely event should be quantified). Therefore, in the risk assessment process it is critical to make the *best educated guesses* possible, and even more important, to *characterise the uncertainty* of those guesses and estimates properly in order to be able to make an adequate prioritisation when implementing the Risk Management plan.

The fundamental difficulty in risk assessment is determining the *rate of occurrence* of an event since statistical information may not be available on all kinds of past incidents. In particular, this is the case when running an exploratory (e. g. R&D) project or operating a new system with novel protocols that are either untested – or tested, but the results are not available for the risk assessment (e. g. in case of corporate competition). On the other hand, it is often difficult to properly evaluate the *severity of the impact or of consequences* as well. This is especially true for immaterial assets where even the scale of measurement is not obvious. To deal with this issue, addressing and providing asset valuations in a proper manner is necessary. Thus, the primary sources of information for risk assessment are so far

- available statistics
- and the best educated opinions.

The main goal of the risk assessment is to present some information to the management of the organisation about the primary risks involved in the underlying venture in such a way that they are easy to understand. This supports Risk Management decisions which can readily be prioritised. There have been several theories and attempts to quantify risks. Among them, the most widely accepted approach is based on the simple finding that states (see Figure 3.3):

Risk equals *Rate of occurrence* multiplied by *Impact of event.*

Definition 3.1.6
[Risk] Risk, R, is characterised by its two measurable attributes, the magnitude of the potential loss, L, and the probability of the event in which the loss occurs, p. It is computed as the product of those attributes

$$R = L \cdot p = Pr(L). \tag{3.1}$$

More generally, if there are several factors that imply the same type of risk, and consequently, there are different attributes that originate in (and lead up to) the same type of risk, the

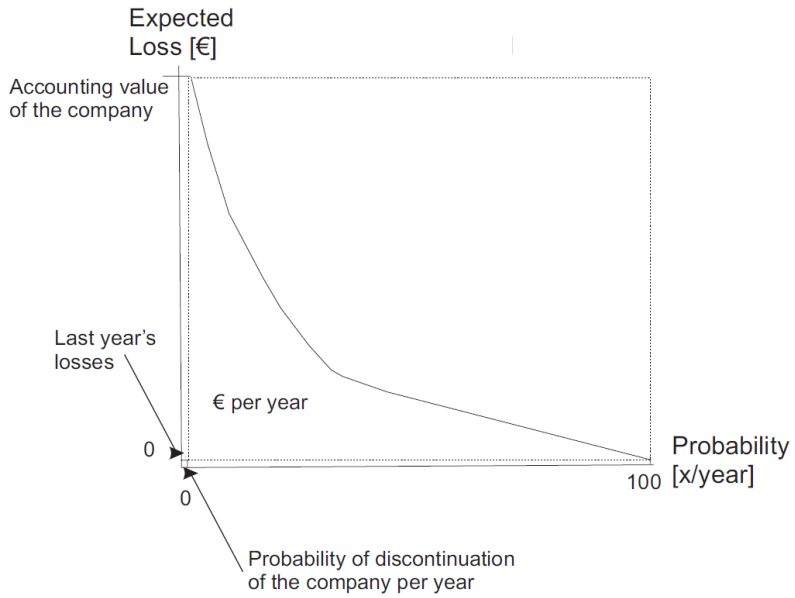


Figure 3.3: Relationship of Expected Loss and Probability of an Event

formula for risk quantification is the following:

$$R_{\text{total}} = \sum_i L_i \cdot p_i \ , \tag{3.2}$$

where L_i denotes the potential losses if an event i takes place with a rate of occurrence $p_i = Pr(L_i)$. The sum is computed on all possible events i which should be incorporated in the risk analysis and imply the same type of risk that is under investigation.

Risk assessment is the most important step in the process of Risk Management. Generally, it is also the most difficult to formulate properly, since in the risk identification factors can be easily neglected or the associated data can be incorrectly presented when their uncertainty should be characterised. After working manually out a proper and convenient formulation, the process of risk assessment reduces to a mechanical exercise that can be carried out by computers.

Note that a risk with a large potential loss and a low probability of occurrence must be treated differently than a risk with a low potential loss and a high likelihood of occurrence. In theory, both can have nearly equal priority to be dealt with first, but in practice it can be very difficult to manage this kind of situation for Risk Management processes handling the scarcity of resources, especially time and computing power.

Lately, several research studies have shown that the financial benefits of Risk Management do not depend much on the formulas used. Instead, the most significant factors in Risk Management turned out to be the following:

1. risk assessment has to be *frequently* performed
   - to avoid decision making with outdated information and

- to be able to work with non-robust situations;

2. Risk Management needs to be carried out by using as *simple* methods as possible
   - to make the managers and responsible persons easily understand the governing principles of the decision making process and (as a consequence)
   - to make them able to update even the whole process of risk assessment if the environment changes.

The following risk assessment foundations are structured as follows: Section 3.1.4.1 gives an overview of basic measures of risk. Since in different fields of application (statistics, finance, etc.) different definitions of risk exist, Section 3.1.4.2 presents these. Section 3.1.4.3 describes how risks are categories into different types. Established risk assessment techniques are pointed out in Section 3.1.4.4.

3.1.4.1 Basic Measures of Risk

Risk is a potentially negative impact to an asset that results in a loss of its value or some of its characteristics. This loss is originated in and generated by either certain unfavourable outcomes of a present process or future events. A main issue here is the correct specification of *loss* as well as the determination of its probability. Here, the difficulties to overcome are due to the following two facts:

1. The sound definition of loss with respect to the framework – it is necessary to recognise and take into account all possible threats that can cause the negative event that is under analysis.
2. The correct quantification of the probability of loss – the rates of occurrence of future events that trigger the negative event and thus impose the associated risk has to be estimated.

In a straightforward manner, the formula for risk can be extended by involving all possible threats and their related losses that imply the same type of risk. 'Decomposing' the risk into elementary factors results in

$$\text{Risk} = \sum_i \Pr\left(\text{Loss}_i\right) \tag{3.3}$$

whereas Loss_i are associated with the risk and none of them can occur simultaneously with any other. This property is called a *disjunctive* decomposition of risk. In particular, the effects of the risk are broken down into n factors, where the potential negative event, if triggered, causes a loss of Loss_i with $Pr(\text{Loss}_i)$ as the chance of happening for a certain index $i = 1, \ldots, n$. Please remark that this relationship actually formulates the additive property of probability measures. The probability of an aggregate of disjoint events is equal to the sum of the probabilities of the individual events.

In general, the overall risk can be computed by using its factors that are not necessarily disjoint. In this case the factors can occur simultaneously, and their implied risk can be

assessed by using the famous *sieve formula* [Halb 74]. For instance, if two factors of some risk are identified, causing losses of Loss_1 and Loss_2, respectively, then the sieve formula reads

$$\text{Risk} = \Pr(\text{Loss}_1) + \Pr(\text{Loss}_2) - \Pr(\text{Loss}_1 \cap \text{Loss}_2) \tag{3.4}$$

Here, $\Pr(\text{Loss}_1 \cap \text{Loss}_2)$ denotes the probability of the losses caused by both factors. On the other hand, in the case three factors with chances of occurring $\Pr(\text{Loss}_1), \Pr(\text{Loss}_2)$, and $\Pr(\text{Loss}_3)$ are considered, where each of them represent the same risk, the sieve formula leads to

$$\begin{aligned} \text{Risk} \quad &= \Pr(\text{Loss}_1) + \Pr(\text{Loss}_2) + \Pr(\text{Loss}_3) - \Pr(\text{Loss}_1 \cap \text{Loss}_2) \\ &-\Pr(\text{Loss}_1 \cap \text{Loss}_3) - \Pr(\text{Loss}_2 \cap \text{Loss}_3) + \Pr(\text{Loss}_1 \cap \text{Loss}_2 \cap \text{Loss}_3) \end{aligned} \tag{3.5}$$

Usually, the probabilities of events are characterised by using statistical methods. Assuming that the events under consideration follow a certain type of *random distribution*, estimators for the parameters that distinguish the particular distribution in its family are derived. By determining the parameters of the particular distribution correctly, the probability distribution governing the events that imply the associated risk are specified. If the sample set of past observations is large enough, the future distribution of the events can be approximated to a high degree of accuracy. The only problem with this approach is that during the development of risk assessment processes the right class of probability distributions have to be found out or assumed correctly. However, in many applications risk assessment experts can draw on the theory of statistics when building the specific methodology for selecting the possible random distributions that can characterise the risk. Some following examples describe this process.

When risks associated with data corruption in telecommunication systems are characterised, the noise that can cause errors in the messages is modelled by so called *white noise*. In general, this means that the risk assessment developers view errors as if they were following a normal distribution - usually denoted by $N(\mu, \sigma)$ – with zero mean, i. e. $\mu = 0$. In this case the *risk* is data corruption, and the *loss* is the cost of correcting the message. The parameter that needs to be estimated to attune the model to a particular situation is the standard deviation σ of the Normal distribution. The larger σ is, the more severe are the errors, and thus the more unreliable the communication is.

On the other hand, when modelling *call systems* and *queuing systems*, the incoming calls/requests are characterised by exponential and Poisson distributions. Calling and queuing systems are the mathematical tools for modelling the overall load and availability of certain systems (providing certain services) that are accessed (usually electronically) by external users. The assumptions of this framework are that

1. a customer calls the system asking some task to be carried out (hence the name call system);
2. the incoming requests of the customers may be put in some order or queue (hence the name queuing system), which, among others, can be based on
   - privilege (distinguishing ordinary/preferred/key customers) or
   - time of request (incorporating the *First-in-First-Out* (FIFO) policy);

3. the incoming requests follow a random distribution, which is defined by a stochastic process.

Assuming that users call the system *independently* of each other, it is clear that when receiving a request, the next request (initiated most likely by another user) will arrive as if the earlier request were not issued. This property is called *memorylessness*, which is only formulated by the continuous exponential and the discrete Poisson distributions. That is, let X denote the *amount of time* that elapses between two consecutive incoming requests, and let Y_t be the *number of requests* that arrive in the time interval $[0, t]$, i. e. between now and t time units where $t > 0$. Then, it is easy to see that

$$\Pr(Y_t = 0) = \Pr(X > t) \tag{3.6}$$
$$\Pr(Y_t \geq 1) = \Pr(X \leq t) \tag{3.7}$$

Using these notations the stochastic processes that govern the call system and queuing system can be mathematically formulated. Acting on the assumption that a request has just been received, the probability that the next request arrives before t time units follows an exponential distribution by

$$\Pr(X \leq t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t} \tag{3.8}$$

whereas $\lambda > 0$ is the parameter of the exponential distribution. When determining Y_t, the challenge is to estimate the parameter of the distribution-class λ as accurately as possible.

Approaches for calling and queueing systems can be successfully applied in a Grid environment. In particular, usage rates of computer servers that provide computing powers to end-users for purchase represent a call system. Moreover, the incoming requests to the computing centres are usually evaluated before a decision is made about its execution. Thus, service providers offering computing power can use models of call systems and queuing systems. The time of receiving the next request and the privilege of its initiator cannot be determined for sure, but rather follows some random distribution that is specific to the environment. The assumption can be followed that end-users – and brokers negotiating on their behalf – act independently, since they all want to optimise their individual position by minimising costs and maximising reliability. However, if their interaction and interference is small, then the assumptions about the memoryless environment (with respect to job and workflow assignment) are fulfilled, and thus the uses of stochastic processes governed by exponential and Poisson distributions are justified. This means that a straightforward application of the call system can be used by service providers to manage their situation and model their relationship with brokers and end-users.

In professional risk assessments, risk includes on the one hand the *probability* of the negative event and on the other hand the *impact* with all its circumstances of that event if it is triggered and comes true. However, if assets are priced by markets or any kind of auction or negotiation-based process, then all probabilities and impacts are reflected in the price. In these cases risk can only come from the variance of the outcomes. This is not an obvious fact, but an important result, which is one of the final conclusions of the Black-Scholes theory for pricing financial assets [Blac 72].

For using risk assessment in the Grid the risk of an SLA violation computed in the Grid fabric is significantly influenced by the resource characteristics. Events influencing or impacting the resources can appear simultaneously even if they are not influenced by each others or have the same cause or origin.

3.1.4.2 Specific Definitions of Risk in Different Fields of Application

There are several definitions of risk, each of which tries to capture the notion of expected loss and combine it with its rate of happening. In essence, they are only different in their specifications that characterise the context of application and the situation where it is used. In general, every risk measure or indicator is proportional to the *expected losses* that is implied by a risky event and the *probability* of that event.

Therefore, the difference between the particular risk definitions depends on the one hand on the *context* of the loss, i. e. how it is defined and specified, and on the other hand on the assessment of the loss which bases on the particular measurement and methodology applied in the task.

In case the losses are clear and invariable, i. e. their associated risks can neither be mitigated nor avoided, then the implementation of risk assessment has to be focused on the *probability* of the event, the *frequency* of the event, as well as the *circumstances* that can trigger the event.

In general, the following two types of risk are distinguished:

1. Risk that is based on scientific and engineering estimations and
2. the so-called effective risk that particularly depends on human risk perception.

In practice, risk assessment techniques that are developed for the analysis of either of these two kinds of risks are in continuous conflicts in social and political sciences. Many informal methods are used to assess or measure risk. Usually, risk cannot be measured directly, instead, some of the factors that are crucial or decisive with respect to the applications and after proper quantification of the data are considered when computing a measure that satisfies some natural and formal properties – which is called measure risk. Applying formal methods the so-called *Value at Risk* (VaR) can be measured [Hoff 02]. In the following certain types of risk definitions are pointed out the rationales behind them are explained.

1. Engineering Definition of Risk:

$$\text{Risk} = \Pr(\text{accident}) \cdot \frac{\text{Costs of consequences}}{\text{per accident}} \tag{3.9}$$

Notice that both terms on the right-hand side are normalised with respect to their respective scaling. The first term is the probability of an event – it is normalised, since it can only take values from the interval $[0, 1]$. Furthermore, the second term represents the relative measure of loss as per accident – the total loss suffered is considered 100%, thus the average loss projected to one accident gives the magnitude of the overall loss in pro-portion to the number of negative events that were actually triggered. Indeed over time the costs are accumulating together with

the number of unfavourable events, however, this is not a problem. The problem arises when costs go up if one accident has happened; this will make the system more expensive, since either the increased risk has to be faced or resources have to be used to mitigate its effects.

2. Statistical Definition of Risk:
In probability theory and statistics, risk is formally identified as the probability of certain events that are viewed undesirable. Usually, the probability of those events and an adequately completed assessment of their expected harm have to be combined in an appropriate (and valid) scenario. This scenario, which represents the outcome of all unfavourable events that can be triggered by the risk, comprises the set of probabilities with their associated *risk*, *regret*, and *reward*.

Using this set of probabilities together with the estimated losses (regrets) and gains (rewards), the expected value of the outcome is assessed as its representative value. Note that this approach is a special case of the calculation of expected utility.

3. Financial Definition of Risk:
This type of risk is specifically defined as the unexpected variability or volatility of returns on financial assets. In particular, its notion includes both potentials; on the one hand the potentially *worse* than expected returns and on the other hand the potentially *better* than expected returns.

These two types of potentials represent the possible reward of having assets that can decrease in value. Indeed, in finance it is understood that higher returns are only achievable if positions are taken that involve higher potential losses. Thus, negative risk (loss) is the price that an investor is willing to pay for the positive risk (gain). This is the price of the opportunity, which is implied and stems from the uncertainty of the opportunity.

4. Definition of Risk in Scenario Analysis:
In the framework of scenario analysis, *risk* is different from *threat*. Here, threat is defined as a serious event with very low probability. In general no reliable statistics for threats are available because either they never occurred or they have only taken place rarely in the past. In particular, no preventive measure against threats exists, by means of which the probability or impact of such possible future events could be reduced, and thus risk assessment methods are actually unable to characterise them. Indeed the probability of a threat is very small; however, the risk assessment is unable to assign any probability to it, no matter what kind of risk assessment methodology is used. In scenario analysis, developers of risk assessment methods are governed by the precautionary principle, and they only seek to reduce threat until it is covered and dominated by other factors represented by a set of well-defined risks. After achieving this goal, the presence of some safety level has been established that hinders the big crashes; thus, the action, project, innovation, or experiment may proceed.

Scenario analysis became the primary risk assessment tool during the Cold War, when the possibility of confrontations between the two superpowers, the USA and the USSR, were an apparent danger. Together with Game Theory, it formed the core of the strategic analysis for both military and political decision making. However, it did not become a wide-spread phenomenon in the fields of insurance until the 1970s, when major oil tanker disasters forced

civil decision makers to develop a more comprehensive foresight. The scientific approach to risk from the viewpoint of scenario analysis entered the areas of finance in the 1980s, when financial derivatives (options and futures) proliferated. However, it did not generally reach most professions until the 1990s, when the power of personal computers reached the level that allowed the collection, management, and digesting of large numerical data collections.

5. Definition of Risk in Information Security:
In the scope of information security, risk is defined as a function of the following three variables:

1. the probability that there is a threat, i. e. deliberate attack,
2. the probability that there are vulnerabilities that can be attacked or exploited, and
3. the potential impact.

If any of these variables approaches zero, the overall risk approaches zero. This fact can serve as a justification for information technology and software company strategies that aim at issuing security updates constantly to the customers (end-users of the systems) that address some recently discovered vulnerabilities (cf. Microsoft).

3.1.4.3 Basic Types of Risk

This section presents the differences of the basic risk types that play central role in risk assessment and analysis in various fields in theory and practice.

Definition 3.1.7
Systematic risk is a source of threat that is implied by a large number of factors and that is triggered by an aggregate effect of several events. From a risk mitigation point of view, it is important to identify this component of the risk which should be analysed. It is important to expose the full extent of systematic risk, no matter what type of Risk Management methods are used since it is impossible to hedge against this type of risk. Introduced in financial applications, the notions of *non-diversifiable risk* and *market risk* are also used for systematic risk.

Systematic risk can only be either *accepted* or (partially) *transferred* – by means of subcontracting. In Grid computing, by default the possibility of some natural disaster, e. g. an earthquake or a severe hailstorm, has to be taken into account that would devastate our facility with the infrastructure, which would make the provider unable to execute the jobs contracted. Another threat of this kind is the chance of getting some political decision – legislation or legislative provision – that would imply difficulties in the providers' everyday contracting procedure by for instance raising barriers in the fields of their core business activity.

Definition 3.1.8
Non-systematic risk is a possible threat that is implied by a small number of factors and that is originated 'locally' with respect to the environment of the problem. Using the same conception known from finance, non-systematic risk is also called *specific risk*. This type of risk represents the main source of *uncertainty* in Risk Management. It is necessary to identify

all kinds of risks and their triggering events, since only these types of risk can be managed by appropriate strategies and policies.

It is readily seen that non-systematic risk can be *transferred*, *mitigated*, or *avoided*. In a Grid environment, both a power failure and the breakdown of a significant number of resources on a Grid site represent non-systematic risks. Obviously, providers can protect themselves against these types of threats (up to some degree) by establishing backup systems and creating contingency plans. The more elaborate the provider's structure is, the better chances it has to finish the contracted jobs even if some non-systematic risks are triggered. However, a more elaborate structure also means higher costs of establishing and maintaining the system. Therefore an important task of the provider is to find the balancing point where the costs and the risks (with accepted systematic risk and mitigated non-systematic risk) are both *low*.

Please remark that a fundamental property of non-systematic risk is that it is usually associated with an entire class that is characterised by some specific feature. In finance, this type of risk is also called a *specific risk*, where it is considered to represent the aggregate (negative) effects of an entire family of assets or liabilities (e. g. tech stocks). The value of an asset, or a collection of assets, may decline over time due to effects that are primarily related to the attributes that the specific class represents. However, this fluctuation can be neutralised by redirecting some of our risk exposures to effects that are not related to this particular type of threat. This is called *diversification* or *hedging*, and in finance it essentially means that additional assets of other kinds to replace some of our original assets are sold or bought. The problem of diversification belongs to the general framework of *Asset Allocation*, which, provided that it is carried out adequately, can protect organisations against non-systematic risk. In principle, diversification is a means of risk reduction (of non-systematic types) because different portions of the market tend to perform and react differently over time. This risk is also referred as *beta risk*.

Diversification is the only way to defend an organisation against non-systematic risk. The following two issues are the most important to address in diversification:

1. to take maximum advantage of the overall (market) conditions and
2. to protect the organisation against specific losses of (market) downturns.

3.1.4.4 Risk Assessment Methods

Usually, the purpose of risk assessment is to find vulnerabilities in a given system, so that they can be corrected. Once vulnerabilities have been identified, it is possible to measure the risk associated with these vulnerabilities [Stew 04]. If the risk assessment model is not formulated properly, for example, because relevant risk factors had been neglected or associated data had been incorrectly used, then the Risk Management process will fail to achieve its objectives. Techniques for risk assessment can be either *quantitative* (i. e. producing a numerical assessment) or *qualitative* (i. e. producing a verbal assessment).

Quantitative risk assessment techniques include:

- Monte-Carlo simulation [Whit 95],
- Fault and Event Tree Analysis [Benn 96],

- Sensitivity Analysis [Whit 95],
- Annual Loss Expectancy [Rain 91],
- Risk Exposure [Boeh 89], and
- Failure Mode and Effects Analysis [Whit 95].

Qualitative risk assessment techniques include [Rain 91]:

- Scenario Analysis and
- Fuzzy Set Theory (FST).

Different risk assessment methods have been identified in the area of software development [Benn 96]. Probabilistic risk analysis is a powerful technique in this application field. *Fault Tree Analysis* (FTA) as well as *Failure Mode and Effects Analysis* (FMEA) are qualitative techniques, which have been used in designing and in testing software products, as well as in the identification of unsafe states [Leve 86]. Software reliability growth models are particularly applicable when the software product is complete and in service.

In [Kans 97], Känsälä developed a quantitative method for risk assessment in software project development. The method supplements traditional software cost models with risk contingency capabilities. The results have been demonstrated with the development of a tool called *RiskTool*.

Risk assessment methods have also been used in the area of e-business development, for example in [Ngai 05], Ngai and Wat used FST and developed a fuzzy decision support system for risk assessment in e-business development.

3.2 Grid

After the Internet was set up in the 1970s-1980s, the access to remote compute resources was realised in scope of Metacomputing. The idea of resource sharing was already pointed out from the Internet pioneer Len Kleinroch two months before he switched on the first node of the ARPANET [Klei 69]. Grid concepts and technologies have been originally used for resource sharing of scientific collaborations. These started with gigabit/sec testbeds [Catl 92, Smar 92]. The first large scale Grid experiment was *Information Wide Area Year* (I-WAY) which connected distributed high-performance compute resources of 17 sites. Authentication of distributed applications and distribution of libraries was realised by a single gateway on each Grid site, the I-WAY *Point of Presence* (I-POP). I-Way was only seen as a testbed in order to prove that distributed access to compute resources is realisable and to identify challenges which have to be solved. After I-WAY, the Grid idea exploited as defined in Chapter 1: resources are not longer limited to be compute nodes, rather file-servers, mobile devices, etc. might be accessed and used from anywhere through a single entry point.

In order to manage the utilisation of and access to Grid resources, *Virtual Organisations* (VOs) are crucial [Fost 00a]. The idea of defining VOs already exist before the Grid [Vyss 65]. Since providers are not willing to allow arbitrary individuals to use their resources, the implementation of VOs forms the basis for authentication mechanisms. The authentication is

realised by certificates which are in control of specific institutes of a VO. Other key functionalities provided by Grid middleware solutions Globus Toolkit [Glob 97, Fost 05b, Globus 08] or Unicore [Unicore 07] comprise resource discovery, monitoring, and security aspects in scope of data transfer [Fost 01]. Note that Globus Toolkit is prevailing and Unicore is used only in a few – mostly European – Grids.

A milestone in the Grid history is the definition of the Open Grid Service Architecture (OGSA) in 2002 [Fost 05a] (see Figure 3.4). OGSA is based on Web technologies since the Grid can be also understood as a collection of services: e. g. a data transfer service provides operations to transfer data from one storage entity/service to another. In this scope service discovery and service composition are important features. Service discovery enables users to find providers for performing a requested action even in environments they are not familiar with. In a service-oriented architecture (SOA) this can be easily realised by service registries. Service compensation is important, in order to ensure the re-usability of existing code and services, and enables to build complex functionalities from them. In addition to OGSA, the *Open Grid Service Infrastructure* (OGSI) [Tuec 03] was developed in order to support basic Grid behaviours. It consists of a set of WDSL interfaces and associated conventions, extensions, and refinements of Web Services standards. A Web Service which conforms to the OGSI standard is a *Grid service*.

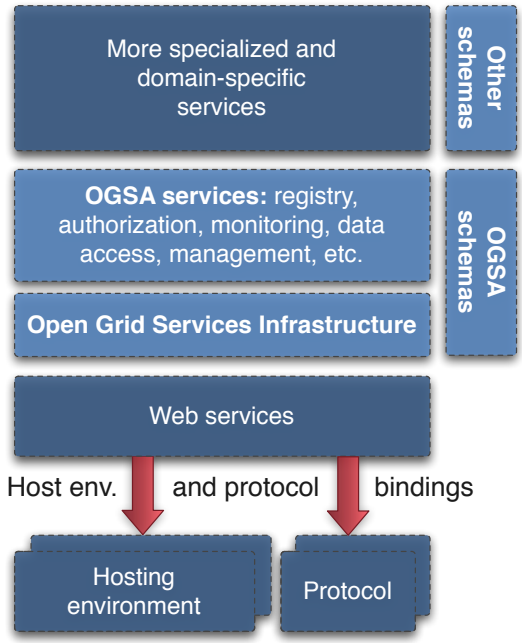


Figure 3.4: Core Elements of OGSA (shaded) [Fost 03]

3.3 Service Level Agreements (SLAs)

Service Level Agreements are used to describe all parameters of a business relationship between service consumers and service providers. The research group *Grid Resource Allocation and*

Acquisition Protocol (GRAAP) of the Open Grid Forum (OGF) [OGF 08] developed the WS-Agreement specification which has been established to define contracts for the usage of Grid services and web services in general including the definition of QoS aspects. Agreements are used in order to define obligations and assurances concerning the delivery of the negotiated service usage.

This section presents the *WS-Agreement* (WS-AG) [Andr 07] specification by describing its content and structure. Section 3.3.1 starts with an overview of service levels followed by the agreement structure according to WS-Agreement in Section 3.3.2. Afterwards the content of an SLA is explained in Sections 3.3.3 and 3.3.4 by following the agreement structure: context and service terms. After the context and organisation of SLAs are clarified, Section 3.3.5 summarises the negotiation processes proposed according to the WS-Agreement specification and the WS-Agreement Negotiation protocol [Andr 06].

3.3.1 Agreements for Service Usage

An agreement specifies the service provisioning between the service consumer and service provider and consequently, it has to contain information about the agreement parties and has to ensure that these are precisely defined. The service consumer is thereby defined usually as the end-user. If a broker is involved in the SLA negotiation, the agreement initiator and the service consumer might differ. In this work no differentiation is made whether the Grid service provider contracts with an end-user or broker. In order to simplify the terminology, the actor the provider is negotiating with is denoted as *contractor*, i. e. it might be an end-user, broker, or another Grid provider. Note that a negotiation of service levels, i. e. agreeing on price and penalty fee for a service request containing guarantees, is provided may be performed between different Grid providers in the scope of outsourcing.

A service agreement is related to a specific service provisioning which has to be defined precisely. The service description must be either part of the agreement's terms or be specified before creating the agreement. Hence, at least either a service reference or a service description element is contained in the agreement. The service description in a computational Grid usually includes the application to be executed, the required number of resources, their usage time, etc.

Requirements for the service usage can be defined by the service consumer reflecting their specific needs and expectations. These requirements are defined by *Service Level Objectives* (SLOs) which reflect the guarantees a provider should give. These SLOs are used to define arbitrary QoS aspects such as the service availability, response time of a service, or a deadline until which the service has to be provided. The definition of QoS aspects within a contract is tightly associated with specifying a compensation if this QoS is not provided. If a service consumer defines multiple QoS aspects, these may be differently urgent to be provided. For example the service availability has the highest prioritisation for the service consumer and a longer service response time is less important. To address this essential point, each SLO can be mapped to a business value which may act as an indicator of importance of this SLO for the service consumer. Contractual penalties can be defined in scope of business values which enables to define multiple penalty payments for not meeting different SLOs.

The agreement creation typically starts with a template which is used to define the particular service contract. The template can contain a set of rules describing the creation constraints of service agreements, i. e. defining acceptable values which might be used in the agreement. For example a creation constraint determines that a job duration must be specified if holding a specific deadline is requested. Note that the constraints provide no guarantee that the service provider will accept the agreement. Since service providers usually advertise the resources and services they can deliver, they define the SLA template which is used by the service consumer to define their SLAs and to initiate a negotiation. After the template has been modified by the agreement initiator according to its service request, it is sent to the agreement responder in form of an agreement *offer*.

If the service usage has a significant influence on the business activities of the service consumer, it is interested to observe the state of the service delivery. An observation enables the service consumer to react early to failures and delays. In order to address these needs, it is possible to monitor the agreement compliance during runtime. In general, the fulfilment of an agreement is crucial for determining whether a contractual penalty has to be paid from the service provider or the service consumer. A service consumer has to pay a contractual penalty if it does not fulfill obligations it is responsible for. Such obligations could be the delivery of input data for a service on time. Usually the contractual penalty of the consumer equals the reward. The agreement compliance can be only evaluated if the service term states are known. Consequently, adequate mechanisms have to be available in order to be able to determine whether all obligations have been fulfilled. The verification of the agreement has a high importance since it determines whether the reward is paid or the provider has to compensate the service consumer's loss by a penalty fee. Since the contractor parties might argue whether the agreement has been fulfilled or not, involving a third party in this process would be beneficial.

The definition of an agreement in the WS-Agreement specification follows in order to complete the overview:

Definition 3.3.1 (Agreement [Andr 07])
An agreement defines a dynamically-established and dynamically managed relationship between parties. The object of this relationship is the delivery of a service by one of the parties within the context of the agreement. The management of this delivery is achieved by agreeing on the respective roles, rights and obligations of the parties. The agreement may specify not only functional properties for identification or creation of the service, but also non-functional properties of the service such as performance or availability. Entities can dynamically establish and manage agreements via web service interfaces.

Note that WS-Agreement depends on other protocols in order to enable specific technical functionalities such as the generation of templates by a factory or the creation and exposing of an agreement. WS-ResourceProperties [Grah 06] is discussed in scope of the *Web Service Resource Framework* (WSRF) and is used for instance to realise the status of the agreement as a resource property. The usage of the WS-ResourceProperties protocol is beneficial since functionalities such as the access to multiple resource properties by sending one request are available. The factory generating agreements returns an *End Point References* (EPR) after creating a new agreement, which is part of the WS-Addressing protocol[5]. WS-

[5] see http://www.w3.org/Submission/ws-addressing/

ResourceLifetime [Srin 06] might be used to destroy not longer used resources if an agreement is not longer needed. WS-BaseFaults [Liu 06] defines the basic faults which might appear when working with agreements.

3.3.2 WS-Agreement – Agreement Structure

In order to be able to map all aspects of a service contract, the WS-Agreement structure can be divided into three main building blocks (see Figure 3.5):

Name The name of the agreement is optional and completely independent from the name of used agreement template(s). The name is not a unique identifier. In order to avoid the handling of the EPR for human-beings, it might be used to provide an understandable name.

Context The context is a required element of an agreement and contains its meta-data which includes the participants and lifetime/duration of the agreement.

Terms The collection of agreement terms describe the contract of the service usage. At least it must contain one *Service Term* (ST). The declaration of several STs and guarantee terms (GTs) is supported but not required.

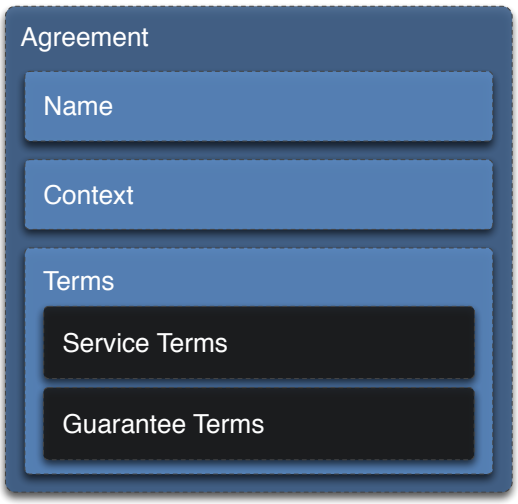


Figure 3.5: Structure of the WS-Agreement [Andr 07]

The XML structure of the agreement according to the WS-Agreement specification is shown in Listing 3.1.

Listing 3.1: Structure of an Agreement according to WS-Agreement

```
<wsag:Agreement AgreementId="xs:string">
    <wsag:Name>
        xs:string
    </wsag:Name> ?
    <wsag:AgreementContext>
        wsag:AgreementContextType
    </wsag:AgreementContext>
    <wsag:Terms>
```

```
        wsag:TermCompositorType
    </wsag:Terms>
</wsag:Agreement>
```

The AgreementIdentifier is mandatory and must be unique between the AgreementInitiator and AgreementResponder. Note that in scope of renegotiations or applying the WS-Agreement Negotiation protocol, a modified agreement document can be considered as an update of an existing agreement. In this context, the updated version has the same name and the same EPR as the original agreement and only the identifier enables the agreement initiator and responder to differentiate both documents. Consequently, when generating a modified agreement, a new identifier has to be specified in order to ensure that both contractual parties are aware which version is currently in use..

3.3.3 WS-Agreement – Context

The meta-data of the agreement is defined in the context. It has to specify the participants of the agreement as well as the expiration time, i. e. this duration of the agreement's validity defines how long the parties are obligated by the terms of the agreement. The definition of such an expiration time is optional and is valid for the whole agreement. Listing 3.2 shows the context in detail:

Listing 3.2: Context of an Agreement according to WS-Agreement

```
<wsag:Context xs:anyAttribute>
<wsag:AgreementInitiator>xs:anyType</wsag:AgreementInitiator> ?
<wsag:AgreementResponder>xs:anyType</wsag:AgreementResponder> ?
<wsag:ServiceProvider>wsag:AgreementRoleType</wsag:ServiceProvider>
<wsag:ExpirationTime>xs:DateTime</wsag:ExpirationTime> ?
<wsag:TemplateId>xs:string</wsag:TemplateId> ?
<wsag:TemplateName>xs:string</wsag:TemplateName> ?
<xs:any/> *
</wsag:Context>
```

The definition of the agreement initiator and responder clarify the roles.

Definition 3.3.2 (Initiator (Agreement Initiator) [Andr 07])
An agreement initiator is a party to an agreement. The initiator creates and manages an agreement on the availability of a service on behalf of either the service consumer or service provider [. . .]

Definition 3.3.3 (Responder (Agreement Responder) [Andr 07])
The agreement responder is a party to an agreement. The responder implements and exposes an agreement on behalf of either the service provider or service consumer [. . .]

AgreementInitiator and AgreementResponder assign the roles to the negotiation parties and may be defined by a URI or an EPR from WS-Addressing. It is also possible to identify the parties by a security identity. The ServiceProvider listed in the context after the responder identifies whether the AgreementInitiator or -Responder is acting as the service provider.

To simplify the validation checks of the agreement responder, the identifier and name of the template are elements of the context. It is important to point out that these elements are generally optional but, if a template has been used, the name and identifier have to be contained. However, these elements are optional.

In order to be customisable for various usages, the agreement allows to define additional child elements and attributes.

3.3.4 WS-Agreement – Agreement Terms

Terms are describing the main content of an agreement. According to the differentiation of STs and GTs, a term is assigned to one of these types. The service terms are defined more precisely by the categories *Service Description Term* (SDT), service reference, and service property. The term definition structure is designed to be able to map arbitrary service and guarantee requirements. In particular the underlying concept is powerful since various combinations of STs and GTs can be defined by using logical connectors: AND by the element `wsag:All`, OR by the element `wsag:OneOrMore`, and XOR by the element `wsag:ExactlyOne`. Listing 3.3 shows the composition possibilities.

Listing 3.3: Agreement Terms according to WS-Agreement

```
<wsag:Terms>
  <wsag:All>
    <wsag:All>
        wsag:TermCompositorType
    </wsag:All> |
    <wsag:OneOrMore>
        wsag:TermCompositorType
    </wsag:OneOrMore> |
    <wsag:ExactlyOne>
        wsag:TermCompositorType
    </wsag:ExactlyOne> |
    {
        <wsag:ServiceDescriptionTerm>
            wsag:ServiceDescriptionTermType
        </wsag:ServiceDescriptionTerm> |
        <wsag:ServiceReference>
            wsag:ServiceReferenceType
        </wsag:ServiceReference> |
        <wsag:ServiceProperties>
            wsag:ServicePropertiesType
        </wsag:ServiceProperties> |
        <wsag:GuaranteeTerm>
            wsag:GuaranteeTermType
        </wsag:GuaranteeTerm>
    } *
  </wsag:All>
</wsag:Terms>
```

Aspects of a service are described by using SDTs, service references and/or service properties. Guarantee terms are optional but can be used to define obligations which are associated with a service term of the agreement. The term definitions are detailed in Section 3.3.4.1 and Section 3.3.4.2.

3.3.4.1 Service Terms

Service terms are either used to describe the service or to define service properties. The other category of service terms - the service reference element - contains a reference, e. g. an EPR, to an existing business service. The most important service term category is the SDT which describes the service about which the agreement is. A SDT element is built of three parts: name of the SDT, the name of the service which is fully or partly described by this SDT, as well as a description of the offered or required functionality. Listing 3.4 shows the XML-structure of SDTs. In the computational Grid SDTs are described by using the *Job Submission Description Language* (JSDL) [Anjo 06] which can be used to specify resources as well as details about data staging. By using the single-program-multiple-data extension , JDSL SPMD [Savv 07], the application requirements are defined. In the WS-Agreement specification it is pointed out that the description element is using a domain-specific language which may be independent of WS-Agreement.

Listing 3.4: Service Description Term of an Agreement according to WS-Agreement

```
<wsag:ServiceDescriptionTerm
    wsag:Name="xs:NCName" wsag:ServiceName="xs:NCName">
    <xsd:any> ...  </xsd:any>
</wsag:ServiceDescriptionTerm>
```

Definition 3.3.4 (Service Description Terms [Andr 07])
Service Description Terms (SDTs) describe the functionality that will be delivered under the agreement. The agreement description may include also other non-functional items referring to the service description terms.

Such non-functional requirements are expressed in the agreement by using variable sets in service properties. An example of a variable set is shown in Listing 3.5 which describes the number of CPUs to be used for the job execution during runtime. Guarantees are defined by using metrics which have to be declared in the `wsag:Metric` attribute. In the example the metric `job:numberOfCPUs` is used which has to be globally defined. The `wsag:Location` element is used to associate the guarantee to a SDT as realised in the example by using XPath. A variable set may contain multiple variable elements, however their names have to be unique within a variable set.

Listing 3.5: Example Guarantee Term of an Agreement according to WS-Agreement

```
<wsag:Variable name="CPUcount" metric="job:numberOfCPUs">
    <wsag:Location>
        //JobDescription/Resources/IndividualCPUCount/Exact
    </wsag:Location>
</wsag:Variable>
```

3.3.4.2 Guarantee Terms

The intention to negotiate service levels is in most cases the request for guarantees and QoS aspects. If those have been accepted by the service provider but could not be delivered as

negotiated, the provider has to pay a contractual penalty in order to compensate for the resulting loss of the service consumer. Guarantee terms describe such assurances associated with the service described in the agreement. In scope of using an agreement for a job submission, assurances are often defined on bounds (minimum/maximum) regarding the resource requirements or time constraints of the service provisioning. The term SLO is used to refer to such bounds. The detailed definitions of both terms in the WS-Agreement specification are following.

Definition 3.3.5 (Guarantee (Guarantee Terms) [Andr 07])
Guarantee terms define the assurance on service quality (or availability) associated with the service described by the service definition terms. They refer to the service description that is the subject of the agreement and define service level objectives (describing for example the quality of service on execution that needs to be met), qualifying conditions (defining for example when those objectives have to be met) and business value expressing the importance of the service level objectives.

Definition 3.3.6 (Service Level Objective (SLO) [Andr 07])
Service Level Objective represents the quality of service aspect of the agreement. Syntactically, it is an assertion over the terms of the agreement as well as such qualities as date and time.

An agreement may contain an arbitrary number of guarantee terms (including none). The structure and content of guarantee terms are shown in Listing 3.6. The SLO expresses the condition which has to be fulfilled in order to satisfy the guarantee.

Listing 3.6: Schema of Guarantee Terms of an Agreement according to WS-Agreement

```
<wsag:GuaranteeTerm Obligated="wsag:ServiceRoleType">
    <wsag:ServiceScope ServiceName="xsd:NCName">
        xsd:any
    </wsag:ServiceScope>*
    <wsag:QualifyingCondition> ... </wsag:QualifyingCondition>?
    <wsag:ServiceLevelObjective> ... </wsag:ServiceLevelObjective>
    <wsag:BusinessValueList> ... </wsag:BusinessValueList>
</wsag:GuaranteeTerm>
```

A guarantee term can be associated with a service scope, i. e. the SDT(s) it relates to. In addition to this, qualifying conditions might belong to the definition of assurance. A qualifying condition is optional and if it is part of the guarantee term it defines a precondition which must be met for a guarantee to be enforced. The qualifying conditions might be set as conditions that a service consumer must fulfil or on external factors.

The importance of a guarantee term from the service consumer's perspective is defined by using business values. These can be also used by the service provider in order to specify the likelihood of meeting that objective. In the commercial Grid business values are used to define the penalty and reward (see Listing 3.7). Hence, the WS-Agreement supports that for each guarantee term individual penalty fees and rewards might be defined. It is even possible to define multiple penalty or reward elements in the business value list. If multiple elements have been defined, these are applied alternatively, depending on the longest assessment interval applicable.

Listing 3.7: Business Value Definition of a Guarantee Term

```
<wsag:BusinessValueList>
    <wsag:Importance> xsd:integer </wsag:Importance>?
    <wsag:Penalty>
        <wsag:AssesmentInterval>
            <wsag:TimeInterval>xsd:duration</wsag:TimeInterval> |
            <wsag:Count>xsd:positiveInteger</wsag:Count>
        </wsag:AssesmentInterval>
        <wsag:ValueUnit>xsd:string</wsag:ValueUnit>?
        <wsag:ValueExpr>xsd:any</wsag:ValueExpr>
    </wsag:Penalty>*
    <wsag:Reward> </wsag:Reward>*
    <wsag:Preference> </wsag:Preference>?
    <wsag:CustomBusinessValue> ... </wsag:CustomBusinessValue>*
</wsag:BusinessValueList>
```

3.3.5 Agreement Negotiation

WS-Agreement is a powerful specification to describe simple as well as complex services and related SLOs. The agreement is defined in most cases based on a template and is sent from the agreement initiator, as an agreement offer, to the agreement responder. The agreement responder decides then to accept or reject the offer.

Definition 3.3.7 (Acceptance (Agreement Acceptance) [Andr 07])
Agreement acceptance is the decision of the agreement responder to participate in an agreement relationship with an initiator as defined in the offer made by the initiator. ...

Definition 3.3.8 (Rejection (Agreement Rejection))
Agreement rejection is the complement of the acceptance process wherein the responder decides not to participate in the agreement relationship described in the initiator's offer.

According to this negotiation workflow an agreement has the following states:

Pending An agreement offer has been made but is neither accepted or rejected.

Observed An agreement offer has been made and accepted. This state might following pending.

Rejected An agreement offer has been made and rejected. This state might follow pending.

Completed An agreement offer has been received and accepted. The completion state is only valid if all activities belonging to the agreement have been finished.

The agreement can only be completed if all services associated with the agreement have been *completed*, i. e. no service is in one of the following states: *not ready*, *ready*, *idle*, or *processing*. The current WS-Agreementspecification intends to define the content of the structure. Previously, it also contained a complex negotiation protocol which was then shifted into the WS-Agreement Negotiation [Andr 06] specification.

Different states are defined in the negotiation process which enables to receive an agreement offer and to answer with a modified version denoted as *counteroffer*. If, for example, the provider is not able to accept a requested deadline, it can send a modified agreement with

a deadline it is able to accept. Another scenario in which an iterative negotiation process is very valuable is that the service consumer defines the SDTs, SLOs, and the penalty fee and the provider determines the reward in a counteroffer.

The state machine describing the negotiation process according to WS-Agreement Negotiation is depicted in Figure 3.6. The state machine differentiates between the state of the agreement initiator and the agreement responder since the agreement is only observed if both parties commit the SLA. In WS-Agreement the agreement initiator always commit the agreement when she has sent it to the agreement responder. A further difference is the start state denoted *advisory*. In this state messages are exchanged between initiator and responder without any effects, i. e. these messages indicate no obligations or restrictions on later negotiation steps. When *soliciting*, the negotiation party (initiator or responder) has no obligations but requires that a counteroffer is accepted. A negotiation party being in the state *committed* implies that the sender accepts the terms if the contract partner moves into the state observed, i. e. also commit the agreement and the negotiation ends successfully. Thus the state *observed* describes that the receiver accepts the agreement to sender has committed to. Note that the state *committed* means that the negotiation party agrees on the obligations defined in the agreement. The negotiation is not successful if it is either *terminated* by using the underlying WSRF termination mechanisms or *rejected* by one negotiation party. The rejection is realised by using the underlying WSRF fault mechanisms to signal the rejection of an offer.

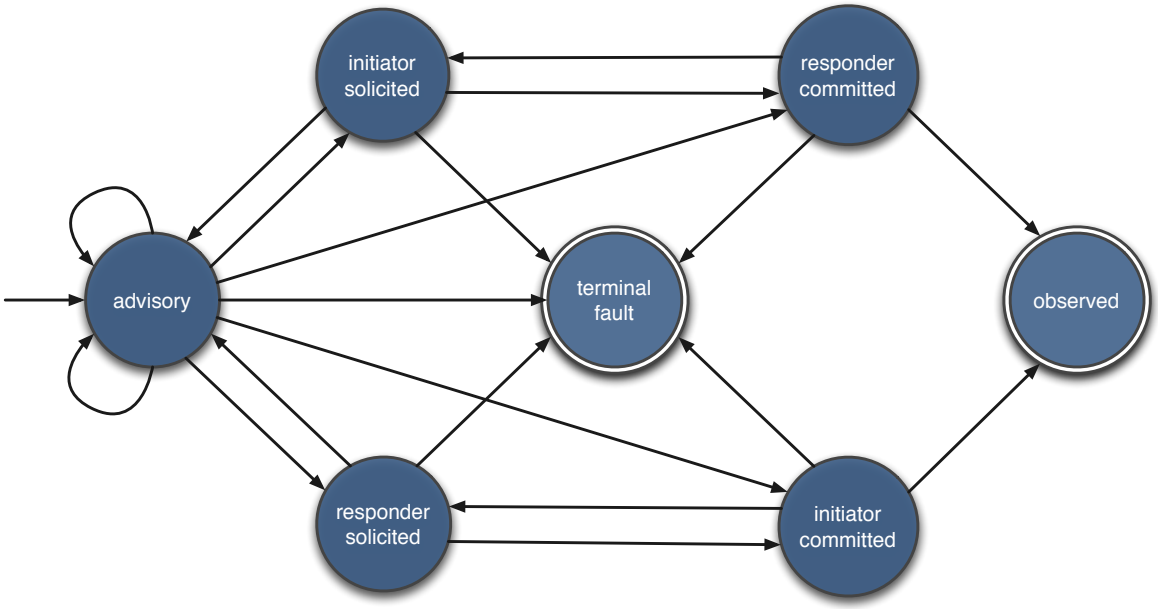


Figure 3.6: WS-Agreement Negotiation - State Machine [Andr 06]

The WS-Agreement Negotiation is significantly more powerful than the acceptance process in WS-Agreement. However, the primary intention of the WS-Agreement specification was to define the content and structure of an agreement. Since until now no implementation of WS-Agreement Negotiation exist but two implementations [Wald 08, Batt 07] have been developed for negotiating agreements with WS-Agreement, current efforts of the GRAAP working group focus on renegotiation of committed agreements [Di M 07a, Di M 07b, Pich 08].

Such renegotiation mechanisms will enable service consumers as well as service providers to ask for modifying an agreement which is in the observed state. Note that initially committing to an empty agreement is also possible and then such a renegotiation equals a negotiation.

3.4 Resource Management in Grids

This section intends to give an overview of resource management in Grids and presents more details than the brief introduction in Section 1.2.3. A RMS has to delivery a set of functionalities in order to support a transparent Grid. Krauter et al. define in [Krau 02] the abstract model depicted in Figure 3.7. They differ between three types of functional units:

Application to RMS Interfaces provides services for end-users and Grid applications. In the abstract structure these are services realising resource discovery, dissemination, brokerage, and interpreting requests.

RMS to native operating system or hardware environment provide functionalities which are used to implement resource management services such as the execution manager or job and resource monitoring.

Internal RMS functions are parts of the resource management service and comprise resource naming, scheduling, resource reservation, and state estimation.

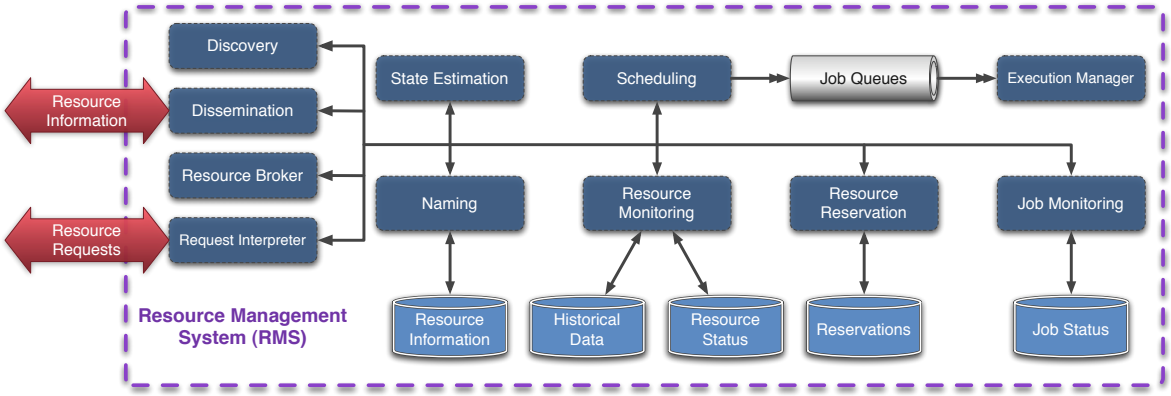


Figure 3.7: Abstract Structure of a Resource Management System [Krau 02]

A further interesting definition of the taxonomy given by Krauter et al. addresses the support of QoS. They have identified that admission control and policies have to be considered in QoS provisioning: "*Admission control determines if the requested level of service can be given and policing ensures that the job does not violate its agreed upon level of service.*" [Krau 02] If users are not able to specify QoS aspects in their request, a RMS does not support QoS at all, even if it would be considered in scheduling mechanisms – which might be count among admission control. Soft QoS support means that a RMS allows end-users to define service levels but it cannot enforce them by policies. Most contemporary Grid solutions support soft QoS for running jobs, however, they lack of ensuring the provisioning of those. Only if all components involved in the job execution can police the service level guarantees, it is in the category of hard QoS support.

Criteria	Queuing Based	Planning Based
planned time frame	present	present and future
reception of new request	insert in queues	replanning
start time known	no	all requests
runtime estimates	not necessary	mandatory
reservations	not possible	yes, trivial
backfilling	optional	yes, implicit
examples	PBS [OpenPBS 08], Loadleveler [IBM 06]	OpenCCS [OpenCCS 08], Maui Scheduler [Maui 08]

Table 3.1: Comparison of Queuing and Planning Based Systems[Hove 03]

In the abstract structure of the RMS (cf. Figure 3.7) the scheduling mechanisms is tightly related with a job queue. As pointed out in Section 1.2.3, RMS have to be differentiated in queuing and planning based ones. Queuing systems usually manage jobs in multiple queues which differ in their policies, priorities, users, etc. If free resources exist and cannot be used by the job at the front of a queue, backfilling [Srin 02, Lifk 95, Talb 99] is often applied in order to execute a job which was not really the next to allocate. Table 3.1 details the differences according to [Hove 03]. Note that Maui can be configured to run as a planning system.

Chapter 4

Requirements for Grid Service Providers

The problem description in Chapter 2 highlighted that *Service Level Agreement* (SLA) provisioning is a challenge for commercial Grid service providers. The applicability of Risk Management to support SLA provisioning was indicated there. Before discussing the details of the Risk Management process, the operation scope of the Risk Management has to be defined. Therefore, the requirements of the Grid Risk Management processes needs to be specified in relation to the objectives of the provider. The results of the requirement analysis are presented in this chapter; Section 4.1 describes the assumptions made and the methods used. Several models can be used to define and specify the requirements. The focus of the requirement analysis is on the definition of a hierarchy of different objectives which is realised by goal models in Section 4.2. Based on the specification of objectives, the resulting functional requirements are presented in Section 4.3.

Further information can be found in [Mold 06].

4.1 Assumptions and Methodology

Requirement analysis can be performed in various ways, ranging from unstructured brainstorming to model based appraisals. The analysis presented here is model based in order to examine the complete field of application. Since Risk Management is a totally different approach from contemporary Grid solutions and developments, the model also takes into account developments which are out of scope of this work. A clear statement of all assumptions ensures that results can be reproduced. The assumptions made are presented in Section 4.1.1. The key aspects of the model which form the basis for the requirement analysis are described in Section 4.1.2.

4.1.1 Assumptions and Dependencies

The integration of Risk Management in the Grid fabric needs to take account of some aspects of the general field of application considered. Firstly it has to be specified whether and which existing technologies or standards are considered. In addition, key aspects of SLA provisioning are essential for the development of adequate Risk Management strategies and risk assessment model. In this section the assumptions of the Grid operation that affect the requirements of an Risk Management process for Grid providers are described. Since this work focuses on

integrating Risk Management for Grid service providers, only constraints related to service provisioning have to be considered. For a risk aware Grid fabric appropriate for in a commercial Grid environment the following constraints have to be satisfied:

1. The design of the system should be integrated within standard Grid architecture, OGSA, and existing middleware implementations such as Globus (see Section 3.2).
2. SLA negotiation is performed using standard protocols such as WS-Agreement [Andr 07] or WS-Agreement Negotiation [Andr 06].

In addition to these design constraints a number of properties have to be considered in the development:

No distinction is made here, between SLA provisioning for simple (single task running on one or serval compute nodes) jobs or workflow jobs. This simplification is valid since from the Grid fabric's perspective these can only differ in the resource requirements since in workflow jobs dependencies between different sub-jobs exist and the output from a sub-job may be used as input from another sub-job. The usage of different resources at different times is also possible for simple jobs. For example a database service may only be needed at the end of the job execution in order to store data.

The *Resource Management System* (RMS) of a Grid provider is assumed to be a planning based system and not queuing based. Since these support advance reservations, planning based systems are better suited to multi-site Grid environments and for negotiating SLAs than queuing based systems are (see Section 1.2.3). The requirements are elicited and analysed in the context of a computational Grid, processing mostly batch jobs that require vast amounts of computational power (see Section 1.2.2). In particular, in the context of service provisioning for compute-intensive applications, risk awareness is crucial in order to estimate the risk of agreeing SLAs and to account for the possibility prevent SLA violations after resource failures. Nevertheless, the results developed in relation to the Risk Management of compute resources will also benefit data Grids that support interactive applications.

The Grid service provider carries out the general Risk Management process. Any fault-tolerance mechanism such as checkpointing, allocation of redundant resources, and migration is assumed to be provided. The possible absence of such fault-tolerance mechanisms does not prevent the Grid provider from supporting Risk Management since it is always possible for a provider to accept the consequences of a risk, restart a job from the initial state, or to outsource part of its workload to other providers.

It is assumed that from the business perspective the end-user is empowered as a customer and is thus enabled to negotiate simple contracts - SLAs - with a broker or a service provider. This assumption reflects the actual business world where contracts and financial decisions are processed at management level. The assumption is made as a simplification since it allows the integration of the risk information in short term contracts that can be processed automated. In this work no further distinction is made between an end-user and the customer.

In summary, the integration of Risk Management should be consistent with standards concerning the Grid architecture and SLA negotiation – OGSA and WS-Agreement or WS-Agreement Negotiation. Furthermore, the focus is on computational Grids in which the highest threat

of an SLA violation is the outage of a compute node. In this context, fault-tolerance mechanisms such as checkpointing and migration are key aspects to support the prevention of SLA violations. However, the absence of these technologies does not affect the applicability of Risk Management since the consideration of risk in the decision making process is still beneficial and a job restart can be performed by any RMS as the most basic fault-tolerance mechanism.

4.1.2 Models Used for Requirement Analysis

In order to define the requirements, the *Knowledge Acquisition in Automated Specification of Software* (KAOS) model [Lams 91] has been used. The KAOS methodology was used to elicit the functional requirements since it allows a clear separation of concerns between the requirements by relating them to distinct objectives. The methodology is based on the progressive refinements of abstract and general objectives into more concrete goals and requirements that lead to the identification of the main entities with their relationships and of the main agents responsible for the satisfaction of the requirements. The methodology is based on the *goal model*, the *object model*, the *agent model*, and the *operation model* as depicted in Figure 4.1.

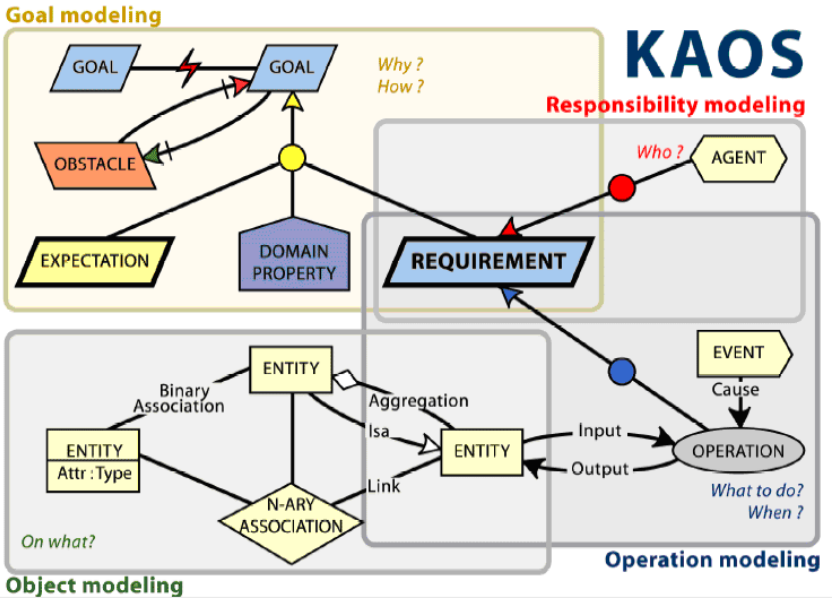


Figure 4.1: Overview of the KAOS Metamodel [Lams 91]

The *goal model* captures higher level and more operational objectives connected by refinement links. Each refinement is interpreted as follows: the parent goal is satisfied if the conjunction of its sub-goals is satisfied. Goals are represented by parallelograms and requirements are represented by parallelograms with a thicker border. A circle connecting the parent goal (indicated by an arrow) to its sub-goals denotes the refinement itself. The result of the refinement process is a goal refinement structure that shows how high-level goals, e.g. strategic goals, are decomposed into lower-level goals that can be assigned to the responsibility of some agent in the system (either a hardware, software, or human agent). Terminal goals

are called requirements and are assigned to some agent in the system. In addition to goals, these diagrams can also capture domain properties (shown as blue "'houses'") which always hold, expectations (shown as yellow parallelograms) which are necessary for the complete achievement of some goal, and obstacles (shown as red parallelograms) which can obstruct the satisfaction of some goals. As goals, obstacles can be refined in order to identify possible causes (vulnerabilities) and reason about the related risk. Obstacles can be addressed using a number of design strategies for elimination, mitigation, and recovery.

The *object model* gives a structured view of the pertinent vocabulary necessary to express the goals. This model is expressed using the classical UML class diagram with entities, relationships, attributes, cardinalities, etc.

The *agent model* gives a view of all the requirements under the responsibility of each agent. This means that each agent instance in the system is responsible for enforcing the requirement instance assigned to him. This model also shows the structure of the interaction among agents in order to cooperate to achieve a higher level goal. The *operation model* is important to assign each requirement by one or several operations performed by the responsible agent.

The focus is on the goal model since it presents the objectives to be followed during SLA provisioning and forms the basis for the developments presented in Chapters 5–9. The object and agent models are derived from the elaboration of the goal model. [Mold 06] presents a more detailed goal model for the complete risk aware framework and all results of the KAOS model, i. e. adequate object and agent models. The operation model is not considered there since its elaboration is best undertaken during specification and architecture designing phases and not as part of the requirement analysis.

4.2 Goal Model Based Requirement Analysis

The introduction of Risk Management into Grid processes enables a wide range of improvements for all three Grid actors: service consumer, broker, and provider. This work focuses on the Risk Management in the Grid fabric since this forms the basis for a risk awareness across all Grid layers. To tap the full potential of risk awareness, enhancements of the processes of service consumers and brokers must not be left out of consideration. Since Risk Management is complementary to current Grid systems, the requirement analysis takes also into account developments which are out of scope of this work. Hence, the analysis starts from a general, high-level perspective and considers aspects and goals which are important for the establishment of risk awareness in Grids pertaining to any Grid actor. Thereby questions arise like: *Do there exist important aspects of consumers of which the Grid service provider should be aware of? What are the main objectives for a Grid service provider?*

At the moment, Grids are primarily utilised in scientific or company-intern contexts. To reinforce the attractiveness and the commercial uptake of Grid services, several issues need to be addressed:

- A first issue is the transparent Grid, i. e. the transparency of Grid services. Consumers of Grid services want to use the Grid transparently without bothering about its underlying

infrastructure. The delivery of Grid services should occur transparently, i. e. without technical overheads.

- A second issue is the trustworthiness of the Grid participants. Consumers of Grid services must feel confident that the *Quality of Service* (QoS) they require will be met in the fulfillment of the contracts agreed with brokers or service providers.
- A third issue is the design of suitable business models that define the relationships between the consumers and the providers in order that demand and offering of Grid services meet in the most efficient way.

These issues are supported by measures such as the use of SLAs, self-management, high utilisation of resources, and fault-tolerance mechanisms. SLAs are the contracts describing the negotiated QoS and their use is a prerequisite and a building block for the wider adoption of Grid services.

Self-management, cost effectiveness, high utilisation, and fault-tolerance all refer to supporting processes that the provider will set up in order to meet the Grid services demand in the most profitable way. The provider has to monitor it resources and to act before SLA violations in order to try to preventing them. Most of the time, the provider will favour the highest possible utilisation rate and if necessary, it will resort to using fault-tolerance mechanisms to ensure the completion of the most critical jobs.

For both, the service provider and the consumer, the commitment to an SLA involves a risk that has to be managed. On the provider side, the Risk Management is related to the management of its resources including the possible fault-tolerance actions and the fixing of suitable prices with respect to the covering of redundancy and margin. On the user side, the Risk Management considers the reliability of the providers concerning SLA fulfillment, their PoF estimations, and the penalties to cover potential SLA violations.

On the basis for these issues, the goal model on Figure 4.2 is proposed. It includes all top level, very general objectives that address the issues mentioned above and that are further refined into underlying objectives which are more specific for the Grid service provider.

The top objective [Attractiveness and Commercial Uptake of Grid Services Improved] is the goal of reinforcing the commercial attractiveness of Grid services. This top objective is the most general one and it is refined into the four high level objectives:

- Grid Services Made Transparent (Invisible Grid)
- Achieve Trustworthiness between all Grid Participants
- Business Models Set Up

The goal [Grid Services Made Transparent (Invisible Grid)] relates to the transparency of Grid services. This is a very important requirement for end-users who want to the details of the underlying Grid infrastructure to be hidden. End-users want to express their needs in terms that are related to their applications and all the details pertaining to the allocation of Grid resources should be hidden to them. Furthermore, end-users should not need to know about mechanisms that may be utilised to prevent SLA violations in the case of resource failures.

The goal [Achieve Trustworthiness between all Grid Participants] relates to the trust relationships that result from an increased dependability and security among the Grid participants.

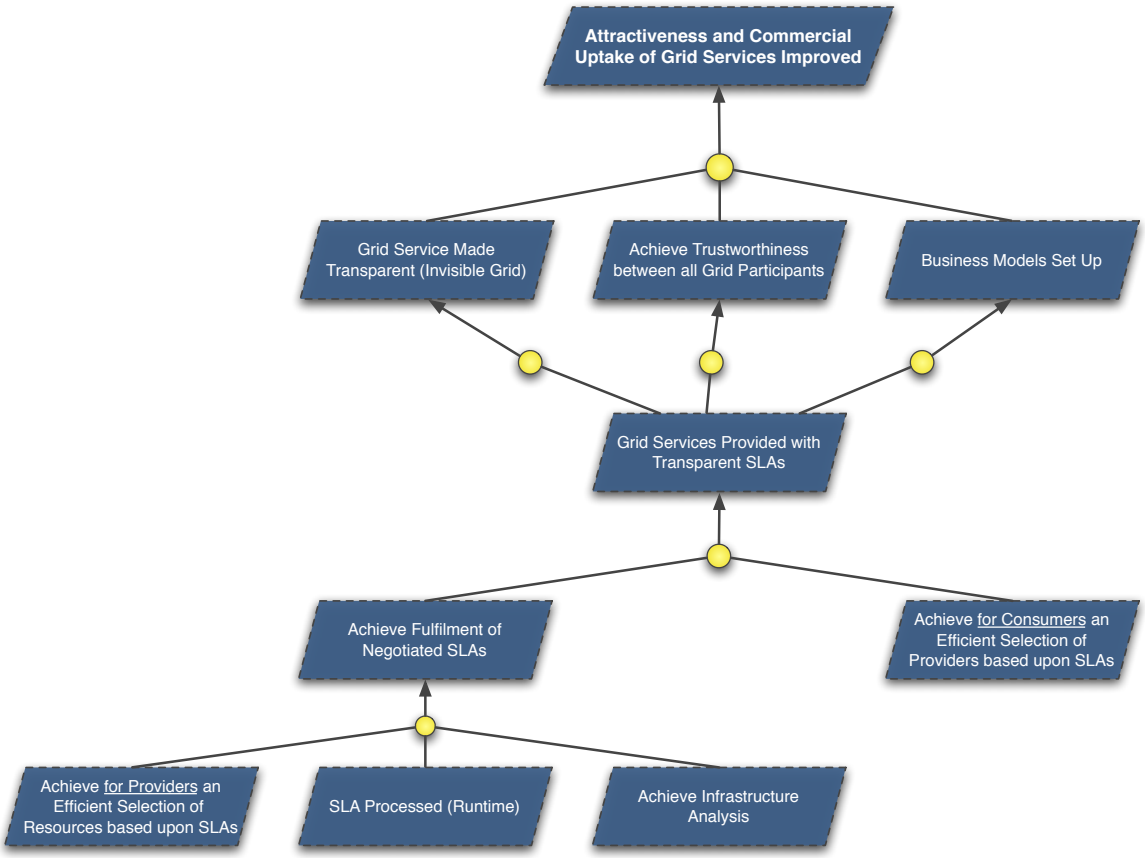


Figure 4.2: High Level Objectives for a Risk Aware Grid

Dependability refers to the fact that end-users can rely on the contracts they agree with brokers and service providers. The dependability of Grid services will be increased if the end-users can build their own opinion about the reputation of the service providers. This can be achieved by maintaining a history of the past activities of the service providers. Security relates to functions such as the identification of customers and providers, the secure transmission techniques, the validation and authorisation of work through the Grid infrastructure. Security is a key element that should contribute to a wider adoption of the Grid services and is beyond the scope of this work. For more information concerning security, [Welc 03b] presents an overview of applied security mechanisms in two successive Globus Toolkit versions. Furthermore the EC-funded project GridTrust [GridTrus 08] is developing a framework compliant with the *Open Grid Service Architecture* (OGSA) [Tali 02] which conforms to a vertical approach for establishing security, privacy, and trust in the Grid. In implementing these, GridTrust's developments have to be coupled with the concept of dynamic virtual organisations, as their state of the art analysis shows [Aziz 07].

The goal [Business Model Set Up] is related to the definition of a suitable business model that allows end-users, brokers and service providers to agree on prices, penalties and risk levels related to the offered IT services. The risk parameters are used as additional parameters in the SLA. Competition between end-users or between brokers to acquire the services is not

considered in the business model. The definition of an adequate business model is out of scope of this work.

The goal [Grid Services Provided with SLAs] on Figure 4.2 introduces the effective notion of SLAs as contracts between service providers, brokers, and end-users. This goal contributes to the realisation of all the preceding high level objectives with the obvious meaning that the SLA is *the key concept* to be considered in the realisation of the high level objectives. The transparency of SLAs does not only refer to the transparency of Grid services as discussed above (see goal [Grid Services Made Transparent (Invisible Grid)]) in Figure 4.2, but also to the fact that the contractors are better able to evaluate the risks they take by committing to these contracts. The risk related to an SLA equals the combination of *Probability of Failure* (PoF) and penalty which are both determined in the SLA. The goal [Grid Services Provided with SLAs] is further refined into the following objectives:

- Achieve for Consumers Efficient Selection of Providers based upon SLAs
- Achieve Fulfillment of Negotiated SLAs

The goal [Achieve for Consumers an Efficient Selection of Providers based upon SLAs] belongs to a service consumer or broker and refers to the selection of a service provider based upon SLA offers. This selection will be carried out by taking into account QoS guarantees offered in the SLA as well as confidence indicators about the service providers maintained by the Grid broker. The provisioning of confidence indicators as an independent evaluation of the reliability of a provider's offered PoF is the main concern of a risk aware broker, details can be found in [Mold 06, Section 3.3.4].

The goal [Achieve Fulfillment of Negotiated SLAs] underlies all the processes of the service provider that enable them to negotiate SLA terms, commit to these terms and take measures to fulfill them. This goal is further refined into the following objectives:

- Achieve for Providers Efficient Selection of Resources based upon SLAs
- SLA Processed (Runtime)
- Achieve Infrastructure Analysis, which is out of scope of this work and details can be found in [Mold 06, Section 3.3.3]

The goal [Achieve for Providers an Efficient Selection of Resources based upon SLAs] is defined as achieving an efficient selection of the resources based on the service consumer's needs as stated in the SLA. It is a key concern of the Grid service provider, which generates the mapping of jobs to suitable resources in order to fulfill the QoS guarantees requested. Since the RMS is planning based, this goal needs to be met at SLA negotiation time and is further refined in the following Section 4.2.1 (Service Provider's Requirements during Negotiation).

The goal [SLA Processed (Runtime)] includes for the provider initiating all necessary actions to fulfill the QoS guarantees defined in the SLA unless it decides to accept the consequences of a failure. This objective needs to be met during the run time of jobs and is detailed in Section 4.2.2 (Service Provider's Requirements in Post-Negotiation).

Starting the goal model on a high-level was necessary in order to tap into the full potential of risk awareness for all Grid participants. The overall objective is to improve the attractiveness

and commercial uptake of the Grid in order to continue the enhancement of the scientific Grid which was initiated by introducing SLAs. Essential for service consumers is the transparency in the Grid which requires that providers hide technical details concerning resource allocation and initiation of fault-tolerance mechanisms in the case of resource outages. Furthermore, cost effectiveness is an essential factor to establish Grid commercialisation for both the service consumer and service provider. Introducing, in addition to price and penalty fees, the declaration of a PoF is one instrument in a risk aware Grid.

The highest level objective from perspective of the Grid provider is to achieve fulfillment of negotiated SLAs since an SLA violation means that a penalty fee must be paid. The realisation of two sub-goals is required to meet this objective. Firstly, an efficient selection of resources during SLA negotiation is important, as described in the following section. Secondly, the provider has to ensure SLA fulfillment after the SLA is agreed as long paying the penalty fee would not be more profitable. Section 4.2.2 presents a finer-grained goal definition for this.

4.2.1 Provider's Requirements during Negotiation

The requirements during SLA negotiation can classified as either general or specific to risk awareness. General requirements identified are valid for arbitrary providers which want to support SLA negotiation. Specific requirements result from the idea of integrating Risk Management processes in the Grid fabric since here, regarding the negotiation phase, the question arises: *How can risk awareness support the decision of agreeing to or rejecting an SLA request?*

The previous goal model of the provider's high-level objectives (depicted in Figure 4.2) has identified that [Achieve Efficient Selection of Resources based upon SLAs] is its primary goal during the negotiation. It is subdivided into five objectives as shown in Figure 4.3:

- SLA Template Advertised, i. e. the provider publishes its template to describe the service offered
- SLA Filled In with User's Needs for the Job, i. e. the provider's contractor (end-user or broker) defines service and guarantee terms in conjunction with a maximum probability of failure (PoF) they are willing to accept
- Risk Aware Reservation for Job Built based upon SLA Request, i. e. the provider has to check feasibility of the SLA according to policies as well as to consider resource and time constraints and failure probabilities before making an advance reservation
- Evaluate SLA request based upon Risk Aware Reservation and Negotiation Policy, i. e. the negotiation module of the provider has to make the final decision about acceptance or rejection by considering price, penalty, and PoF
- SLA Negotiation Completed, i. e. reaching mutual commitment of both contracting parties

The requirements identified are partially valid also for providers *not considering risk*, as long as they support SLA negotiations based on one of the standard protocols WS-Agreement or WS-Agreement Negotiation. Each provider has to advertise its template, ensure that the SLA is filled with the user's requirements, evaluate an SLA request, and complete the SLA negotiation. Two extensions resulting from risk awareness exist, which are crucial to achieve

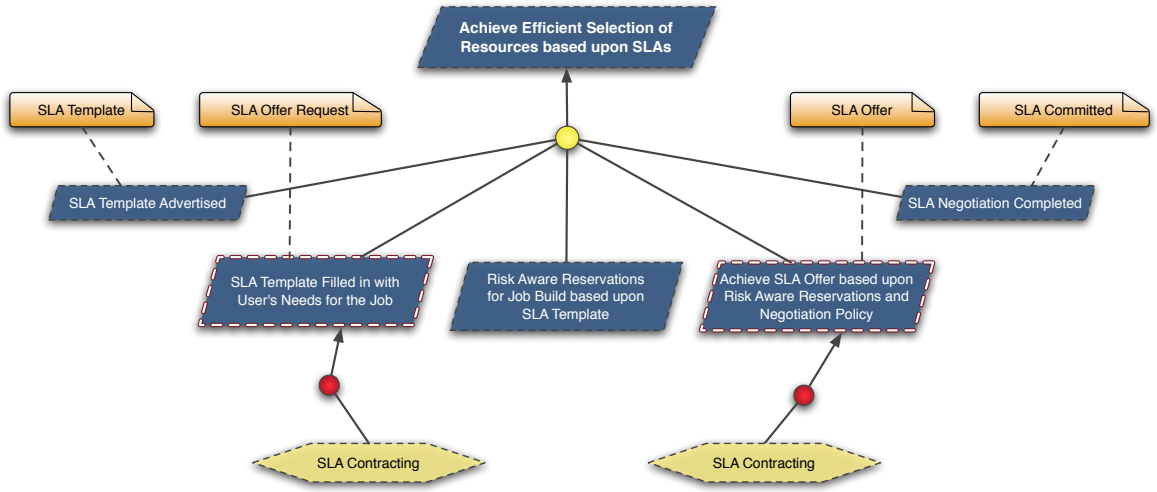


Figure 4.3: Achieve Efficient Selection of Resources based upon SLAs

trustworthiness between all Grid participants and cost effectiveness of Grid services, previously defined as high-level goals. Firstly, making an advance reservation has to result in an estimated PoF for the SLA. Secondly, the estimated PoF has to be compared with the maximum the consumer is willing to accept[1] and acts as a decisive factor regarding acceptance or rejection for the negotiation module.

4.2.2 Provider's Requirements in Post-Negotiation

Similar to the negotiation phase, some requirements identified for SLA provisioning in the post-negotiation phase are generally valid and not specific to risk aware service provisioning. However, introducing Risk Management into the Grid fabric, enables the provider to use risk as a key decision-making factor. In this scope, the following questions can be answered: *If an SLA has been agreed, can risk awareness assist in the prevent SLA violations? What is the impact of risk awareness on the profit?*

In the high level goal analysis (see Figure 4.2) the goal [SLA Processed (Run Time)] has been identified. In addition to the selection of resources during the negotiation, it conforms to the primary goal for the provider in the post-negotiation phase. It is further refined into the following objectives as shown in Figure 4.4:

- Negotiation Completed
- Achieve Management of Precautionary FT-Mechanisms
- Achieve Management of Failure
- SLA Completion Reported

The goal [Negotiation Completed] is a requirement under the responsibility of the scheduling process and depends on successfully making a suitable advance reservation and achieving a mutual commitment of both consumer and provider.

[1] if the provider is not truth-telling, the estimated PoF is compared according to a policy

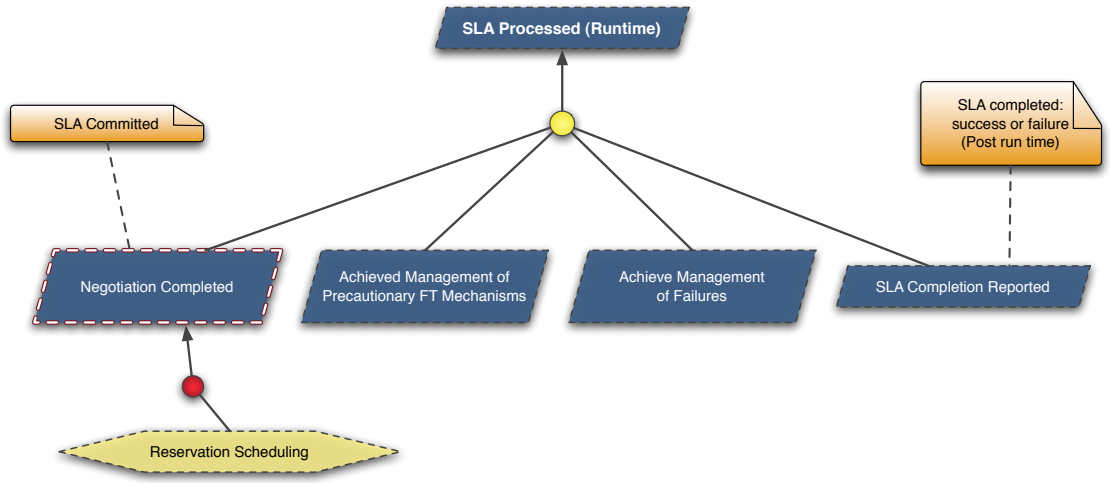


Figure 4.4: SLA Fulfilled at Run-Time

The goal [Achieve Management of Precautionary FT-Mechanisms] refers to the management of those *Fault-Tolerance* (FT)-mechanisms that are initiated as a precautionary measure, based upon dynamic changes to the estimated PoF or the system. It has the sub-goals:

- Failure Probability Updated with Dynamic Change, i. e. based on monitored events the validity of the initially estimated PoF will be controlled and if necessary modified
- FT-Mechanisms Launched, i. e. if the PoF has changed, the RMS evaluates whether to initiate one of the FT-mechanism supported by considering expected profits and losses

The goal [Achieve Management of Failures] refers to the management of FT-mechanisms in the case of a resource outage which would lead to the failure to complete a job which is already executing. A cost-benefit equation is thereby essential for a commercial provider.

The goal [SLA Completion Reported] is the objective of notifying that the SLA bound job has been completed with the consequence that the SLA is marked either as successful or failed in the case there has been a violation of any QoS guarantee term included in the SLA.

The main goals for SLA provisioning in the post-negotiation phase are related to the prevention of an SLA violation. Consequently, risk can be used as a decisive factor in the management and initiation of FT-mechanisms which are either launched as a precautionary measure or after a failure. Since in all decisions the profit will be taken into account, the risk awareness will not be unprofitable, rather it will enable providers to select those job to be resumed whose execution should result in the highest profit.

4.3 Functional Requirement Description

The goal model analysis pointed out the objectives which should be achieved from the provider for a successful and profitable SLA provisioning. The required capabilities in the RMS can be derived directly from these goals. This section summarises the functional requirements identified on the basis of the goal model.

The key idea of this work for achieving the overall high-level goal to improve attractiveness and commercial uptake of Grid services is the integration of Risk Management into Grid processes. A reliable job execution forms the basis for the establishment and acceptance of the Grid in a commercial environment. Hence, enhancing the processes of Grid providers by risk awareness should have significant effects for all Grid participants. In this context the purpose of a Grid provider is to offer risk aware transparent usage of Grid resources for contractors (brokers or end-users) by hiding the complexity of the underlying infrastructure and processes. Since SLAs are a key concept for Grid commercialisation, these are assumed to be used to contractually define the performance and QoS negotiated between providers and contractors. In order to execute SLA bound jobs, the provider must be able to:

- Negotiate SLAs with contractors based on pricing and customer policies as well as risk awareness.
- Accept jobs from contractors.
- Monitor job execution.
- Transfer results to contractors.

An SLA agreement is a business risk for providers since they have to pay a penalty fee if an SLA is violated. This work integrates risk assessment and management into the Grid fabric in order to form the basis for providing SLA bound jobs. This ensures that the business risk for agreeing an SLA can be calculated and based on this information a provider can agree or reject an SLA request. For a risk aware job execution with negotiated SLAs the provider should be capable of:

- Creating SLA templates for advertising the provider's capabilities.
- Receiving SLA requests (from contractors).
- Computing the overall probability of failure of an SLA.
- Considering the availability, plan, and evaluating the effects of fault-tolerance actions (e. g. redundantly executing jobs, checkpointing, internal migration, or outsourcing) according to the negotiated PoF, penalty etc.
- Creating risk aware schedules and making advance reservations based on policies and the upper bound for PoF accepted from the contractor
- Agreeing SLAs.
- Reacting to problems with Risk Management after an SLA has been agreed.
- Evaluating problems after the job execution to detect failure sources in order to prevent failures in the future.

The requirements listed above ensure SLA provisioning in general as well as the applicability of Risk Management processes. Using risk as a decisive factor in the negotiation, scheduling, and planning of fault-tolerance mechanisms is part of Risk Management and requires the estimation of failure probabilities. All requirements can be mapped to goals identified during the requirements analysis. Beyond requirements addressing the general SLA negotiation processes, most issues listed belong to the negotiation phase and thereby to the goal [Achieve Efficient Selection of Resources based upon SLAs]. Reacting to problems with Risk Management is derived from the goal [SLA Processed (Run Time)].

Chapter 5

Risk Management in the Grid

In order to establish the use of SLAs in the a commercial Grid environment, providers need to be supported with information that describes the risk they would be taking by accepting an SLA. In addition, a provider's resource and job management has to balance competing service requests and find the most profitable solution regarding those SLAs, which have been already accepted. If during SLA negotiation conflicts or concurrency between the new received SLA offer and SLAs accepted occur, the SLA offer is rejected. Thus, balancing competing service requests can be performed easily during SLA negotiation. In contrast to this, it is a difficult task when managing resource failures since several SLA bound jobs require use of the same resources. *Fault-tolerance* (FT)-mechanisms have been developed to mitigate the problems caused by resource failures. If many outages occurred, it may be that not enough free alternative resources are available. In contemporary *Resource Management Systems* (RMSs) such situations imply that the job(s) affected by the outage will not be resumed. In rare situations this might in expectation result in the lowest loss, if a job with the lowest priority was affected (i. e. a job having less strict *Quality of Service* (QoS) restrictions, lower penalty fee, and little cost already spent for the job execution in comparison to other jobs in the system). However it is more likely that a job being not of the lowest-priority is affected and violating this job would result in a higher loss.

The key idea to build a framework for making commercial controlled decisions in the RMS is to integrate Risk Management. Therefore, the classical standard Risk Management workflows have to be modified in order to apply Risk Management in the Grid. This chapter presents and justifies the modifications that support the Grid Risk Management process and forms the basis for the integration of risk awareness in the Grid. In order to develop Risk Management activities for the provider, the underlying model of the RMS has to be specified. Chapter 6 details the RMS model assumed and describes the risk assessment method used. The risk aware decision making of the provider is categorised according to the status of the SLA negotiation: Risk Management during the SLA negotiation is presented in Chapter 7 and Risk Management in the post-negotiation phase is described in Chapter 8.

In order to avoid the development of a specific solution and ensure practical suitability, the usage of a standard Risk Management process would be beneficial. The applicability of a standard process is discussed in Section 5.1. This section explains that a standard is not applicable in particular because of the human interaction, and consequently the Grid Risk Management process has been developed. Its description starts in Section 5.2 in which the main configuration and Risk Management tasks to be addressed for Grid usage are *written*

in italics. In order to exploit the full potential of risk awareness in the Grid, a modification of the general Grid Risk Management process can be used which is targeted on a specific decision. Section 5.3 presents this process and completes the framework for integrating Risk Management in the Grid. The main aspects identified in the definition of the Grid Risk Management process are analysed in Section 5.4 by considering the question of whether they can be generally defined for providers. If a general definition is not possible, it is clarified in which context they can be defined. Section 5.5 briefly summarises this chapter.

5.1 Applicability of Standard Risk Management Processes in the Grid

Standards for Risk Management processes have been established as described in Section 3.1.2. These standards, FERMA [FERMA 03] and AS/NZS [ASNZS 99], are applied in companies to define Risk Management strategies addressing activities belonging to their main business tasks. In order to justify the need for a specific solution for the Grid, it is necessary to point out the reasons for this decision. Furthermore, the question of why FERMA was suitable to be modified for a Grid Risk Management process has to be answered.

The FERMA standard (described in Section 3.1.2) defines more precisely which tasks have to be performed in the scope of Risk Management. According to FERMA, first of all the strategic objectives of a company are analysed. According to these objectives the main risks can be identified which represent threats to the achievements of these objectives and which could result in losing profit. The next step is to develop plans which are initiated if those risky events occur in addition to determining what can be done in order to reduce their likelihood. The plans defined consider risks and suitable countermeasures. The decision is taken by responsible staff whose assignment is determined in the definition of the Risk Management. Risk Management processes for different organisations usually differ significantly because of the complexity of a Risk Management strategy addressing the whole business of a company. Additionally, the behaviour of human beings needs to be considered and plans developed in order to prevent their actions negatively impacts on the achievement of one of the main strategic objectives.

Risk Management processes in the Grid for supporting SLA provisioning are very similar for all providers. Hence, a developed Grid Risk Management process can be integrated in various RMSs. By defining appropriate policies and rules, the provider's specific requirements and objectives can be mapped. However, Risk Management processes that are integrated in the Grid, have to be completely automated capable of running without any human interaction. Then consequence of this requirement is that a standard Risk Management process without modifications cannot be applied in the Grid. When multiple events are managed in parallel in standard Risk Management processes, these events usually differ in their types, e. g. a Risk Management process is performed since a supplier could not deliver goods on time and another Risk Management process handles the issue that one of the self-operating production engine has broken. The Grid Risk Management process is focused on the main business of a commercial Grid provider which is the execution of Grid jobs and in particular in a commercial Grid environment the provisioning of SLAs. Thus the same decisions have to be made for different jobs by considering the same issues and questions. Furthermore, all

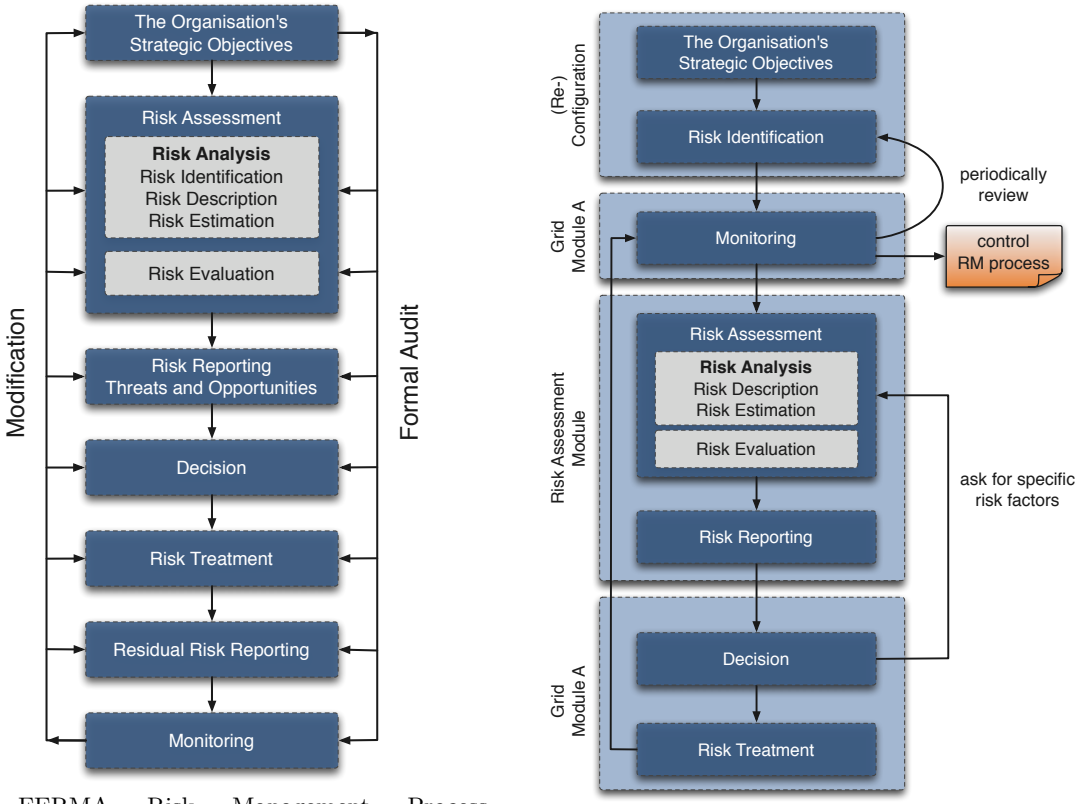
information used during the risk assessment can be automated collected in the Grid. Hence, the monitoring of data is a more powerful mechanisms in the Risk Management process than it is in standard Risk Management processes. This advantage should be reflected in the Grid Risk Management process.

The Grid Risk Management process has to be automated and run without any human interaction. Only the configuration phase may require interaction with an expert, for example, with a system administrator. The FERMA standard process requires human interaction particularly since the risk identification phase includes a selection of risks to be addressed by the Risk Management process. Accordingly, the Grid Risk Management process has to be modified to run without human interaction. Since the main steps in the FERMA standard can be mapped to appropriate tasks in the Grid, this standard has been used as a starting point. Note that the AS/NZS standard could also be modified for the Grid, however, the FERMA standard is more detailed and its task definition is suitable in the Grid context.

It has to be pointed out that the FERMA Risk Management process describes a standard process applicable to a wide variety of organisations to perform Risk Management for arbitrary workflows. In this work the Grid Risk Management process should only focus on SLA provisioning instead of Risk Management activities of the provider which are decoupled from the job execution in order to develop an automated software process. E.g. the Grid Risk Management process should not consider problems such as that new hardware components not being supplied on time or legal provision concerning the provider's staff employment not being fulfilled. As a consequence, the FERMA standard support the consideration of many functionalities and aspects which have to be re-focused in order to apply it in the Grid. In most cases the steps can be termed as in the FERMA standard, but the associated activities are more targeted in the Grid Risk Management process. In order to describe the modifications applied on the FERMA standard, the general idea of the Grid Risk Management process is outlined in the next section as well as detailing the steps. The task description refers to the FERMA definition in order to point out similarities and differences.

5.2 Steps of a Grid Risk Management Process

Risk Management processes can be divided into two main parts: the risk assessment and the decision making procedure which initiates reactions based on the risks assessed. The main task of a Grid provider's risk assessment is to compute accurate probabilities of threats based on a sufficient set of input data provided by a monitoring system. A value which is often assessed is the *Probability of Failure* (PoF) for executing a job defined through an SLA on one or several pre-selected resource(s). In the assessment monitoring information of the pre-selected resources, such as their average uptime – i. e. *Mean Time Between Failure* (MTBF) –, as well as the overall system utilisation for the job execution should be considered. The system utilisation of suitable resources is an important input to enable an evaluation of the number of free alternative resources which might be useable in case of a resource outage. The decision process for the initiation of a countermeasure relies on the functionalities of the RMS as well as the underlying Grid infrastructure. As is usual in Risk Management processes, also the decision process in the Grid takes account of both the success probabilities and the negative consequences (often in terms of cost) for performing a possible Risk Management activity.



(a) FERMA Risk Management Process [FERMA 03]

(b) The General Grid Risk Management Process

Figure 5.1: FERMA Risk Management Process and Grid Risk Management Process Derived from FERMA.

The *risk assessment* requires an analysis of problems and criteria which should be considered in the Risk Management process. The first step of the FERMA standard (repeatedly depicted in Figure 5.1a) and the derived Grid process (see Figure 5.1b) establishes the context of the Risk Management. The different tasks are defined for each step in Section 5.2.1 – 5.2.7 and summarised in Section 5.2.8. These results have been published in [Voss 07b].

5.2.1 Defining Strategic Objectives

The organisation's strategic objectives have to be considered in order to define the risk assessment and treatment. As a result, criteria for the risk assessment are specified in the first phase. These criteria have to be selected carefully since they have implications for the goals of the risk assessment. In business and industry the defined goals should not be too presumptuous in order to not discouraging the responsible staff [Koll 99, p. 14]. However, too low a level is also not advantageous because otherwise the necessity for and the benefits of the risk assessment and management are not clarified.

In the Grid context, the objectives for the risk assessment are layer-specific (Grid middleware and Grid fabric) but almost the same for all organisations: the Grid middleware broker

aims to optimise the mapping from customer requirements and providers' resource offerings; the Grid resource provider is interested in maximising profit by delivering a high quality service, having a high system utilisation, and successfully fulfilling negotiated SLAs. Hence, the coarse-grained strategic objective of a layer-specific risk assessment is defined. *A detailed specification of strategic objectives* can be carried out specifically for each layer. According to the FERMA standard the first step contains the company specific configuration of the Risk Management by defining policies which specify customer or system priorities, or risks which will be accepted. The realisation of this includes the *definition of applicable policies* that conform to Risk Management features. Section 5.4 details the context in which policies have to be defined and which decisions they might influence. In order to support the integration of the developed Grid Risk Management process for any Grid actor, they can adjust these policies and add further ones during the initial configuration phase.

5.2.2 Risk Identification

After the definition of the strategic objectives, the *risk identification* is performed. This task aims to identify an organisation's exposure to uncertainty and is consequently an important issue in the process of Risk Management. It is based on and carried out after the Risk Management context has been determined. If the issues of risk identification are adequately presented and addressed in the identified problems, the calculations and interpretations associated with risk assessment become more convenient and reliable. In general, risk identification is an essential connective link of the Risk Management framework: it is crucial to identify the relevant risks before assessing them. The risk identification process describes and defines threats/problems, their origin, and their consequences. In the FERMA standard this step includes the consideration of the market in which the organisation operates, as well as the legal, social, political and cultural environment in which the organisation exists. This can be simplified for the Grid Risk Management process since those circumstances are similar for all Grid providers and the Grid Risk Management focuses on managing SLA provisioning rather than all threats to and risks for the organisation. Seen from this angle, a threat in the computational Grid is, for example, a compute node outage. Its impact or consequence is that jobs scheduled to run on that resource cannot be executed there (on time). The origin or source of the resource outage is often hard to identify: it could result from a too high CPU temperature caused by a fan defect or insufficient cooling, a software error, network connection error, etc. Crucial information for performing the risk identification concerns hardware characteristics and their consequences on the resource availability. This is, however, very system specific. Adequate information can be collected by monitoring systems, defined by experts, or read out from external databases which provide, for example, the MTBF or the probability of a resource outage caused by a CPU temperature higher than 64 degrees.

In the Grid Risk Management process the risk identification is initially done by an expert during the configuration phase since the analysis may be very system specific. Default results of the risk identification performed in scope of this work can be used by experts as a basis for their own analysis. One of the key differences between the FERMA standard and the Grid Risk Management process is that the risk identification is decoupled from the risk assessment in the Grid Risk Management process. This movement is executed directly after the definition of the objectives. As a consequence, the risk assessment can be performed without any interaction

with human beings. Further, it can be executed in parallel for similar estimations, which is important in the Grid context since it might be necessary to compute the failure probabilities for different SLA bound jobs at the same time. Due to its new positioning the risk identification is only performed during the configuration phase in the Grid Risk Management process. In order to provide an adaptable framework, the thresholds of events and their influence in the risk assessment can be adjusted by running *periodical updates*. The information considered for an automated modification of the risk assessment is very important and has to be well-defined for the resources involved in the Grid. Since various Risk Management processes will run in parallel in the Grid system, only a periodic execution of the risk identification is necessary. This is another justification for extracting the risk identification from the risk analysis and risk assessment as defined in FERMA.

5.2.3 Monitoring

In FERMA the term *monitoring* describes the observation of the Risk Management process and of activities initiated in order to identify aspects/mechanisms which should be improved or adjusted according to dynamic changes of the organisation's objectives or within workflow executions. Since such modifications of the risk assessment or the risk treatment are not realisable in an automated manner, the observation of the Risk Management process has to be decoupled from the Grid Risk Management process. Due to the periodically performed risk identification process the adjustment of thresholds used in the risk assessment is still part of the Grid Risk Management process. A complete observation has to be performed manually in order to identify whether further or additional risk treatment methods should be integrated. It is noteworthy that a change in the organisational's objectives of the provider, which has significant effects on the Grid Risk Management process, is improbable since the Grid Risk Management process is focused on SLA provisioning and in this context objectives are not likely to significantly change. To modify the Grid Risk Management process according to such slightly changes, the automated Grid Risk Management process can be adjusted by re-configuring appropriate policies.

The monitoring step of the FERMA standard can be used in the Grid Risk Management process, making used of traditional monitoring systems used in the Grid fabric and cluster environments, like Nagios [Bart 05], Ganglia [Mass 03], etc. According to this approach, monitoring data in the Grid Risk Management process refers to the information collected from monitoring systems that are integrated as software components in the Grid fabric. The *minimum required information set*, which can be automated read out from monitoring systems and external data sources, have to be declared in conjunction with its *information sources* to define the monitoring's main task and responsibility. In contrast to the FERMA standard the monitoring is displaced directly after the configuration phase instead of running it as the last step of the Risk Management. The repositioning is justified by its modified task in the Grid Risk Management.

5.2.4 Risk Assessment

The risk assessment process is divided into two main sub-tasks: risk analysis and risk evaluation which are described in the following sub-sections.

5.2.4.1 Risk Analysis

The FERMA standard splits the *risk analysis* into *risk identification*, *risk description*, as well as *risk estimation* and aims to identify and compute various risk factors based on the available input data and desired complexity. The risk identification is extracted from the risk analysis in the Grid Risk Management process as discussed in Section 5.2.2. The risk description task is defined in FERMA as producing a structured overview of risks. Therefore in classical Risk Management processes tables are often used which show the risky events, probabilities, consequences, and possible countermeasures. In an automated Risk Management process it is not necessary to develop such a structured overview, e. g. in a table, since all information is evaluated by other software modules. The general methods to manage a risky event are clearly defined in the context of SLA provisioning and consequently, this information does not need to be evaluated during the risk description.

The estimated probabilities of the occurrence of an event/threat are the essential results of the risk description. Since the probability presentation is assigned to the risk description task, it has to be part of the automated Risk Management steps and must not be displaced in the configuration as the risk identification is. Defining the methods used for computing the probabilities is seen as a separate task from the automated Risk Management process. Further, a *precise input specification for the different risk factors* is performed manually in the configuration phase of the company's individual Risk Management process. Dependent on the underlying risk assessment model, a definition of *required and optional input*, which will influence a risk factor, might be important.

The *risk estimation* is the last step of risk analysis in FERMA and evaluates the importance of a risk. The evaluation considers the probabilities of occurrence along with the expected consequences. Hence, after performing the risk estimation, risks are classified by linguistic attributes such as *low*, *medium*, *high*. The risk estimation has to be performed in the Grid Risk Management process exactly in the same manner as in FERMA. This forms the basis for the process of handling the various identified risks.

5.2.4.2 Risk Evaluation

Risk evaluation is the last step in the scope of the risk assessment and compares results of the risk analysis with those criteria and values which have been defined in the context of evaluating the organisation's objectives (see Section 5.2.1). FERMA mentions as criteria, for example, cost and tasks to be performed in addition to legal provisions. The risk evaluation supports the process of determining the relevance of a particular risk for the organisation as well as whether to accept a specific risk and its treatment. The comparison of the assessed risks with the organisation's specific criteria may be performed in the Grid Risk Management process with configured policies. A ranking of risks is built and negligible risks (which are too low/unimportant) will not be considered in the following steps of the Risk Management. Thus, for the evaluation process, the risk assessment module requires *policies to enable it to compare different risks and to decide whether a risk is negligible*. In order to support SLA provisioning, thresholds for non-negligible risks can be defined. Note that it is meaningful to define such thresholds in comparison with the PoF which was assessed during the negotiation. In addition to these probability thresholds, the provider can define thresholds (in money) which determine

the maximum penalty the provider accepts. Since risk is in professional Risk Management the product of the probability of an event and the expected loss of that event [ISO 02], the unit of a risk of paying a penalty fee is expressed in money.

5.2.5 Risk Reporting

The next step after the risk assessment in both the FERMA and the Grid Risk Management process is *risk reporting* which is responsible for publishing information. Since FERMA considers global Risk Management processes for all tasks and workflows in an organisation, the risk reporting differs between internal and external notifications. Internal reporting has to take account of the fact that different levels within an organisation need a different view of the risk assessment's results: the board of directors are interested in the most significant risks, how a crisis will be managed, and whether the process can be performed effectively; business units should only be notified about risks which are related to their working field; individuals have to be notified about the existence of Risk Management and their accountability for specific risks. External reporting is necessary since stakeholders are interested in an organisation's internal management.

The Grid Risk Management process is focused on SLA provisioning and an automated execution; accordingly, the risk reporting only has to notify involved software components. Since the decision making process is integrated in other Grid modules, such as in the scheduler of the provider's RMS, the assessed values have to be *published* from the software component responsible for risk assessment. At this stage it is important to provide consumer-oriented information publishing. Not every Grid module should be aware of every risk factor. A Grid module should only be notified if the risk factor was evaluated as not negligible and the module takes this risk information into account in internal processes. *Filtering risk information* to notify only responsible modules improves the efficiency and enables the integration of authorisation mechanisms. In the development process of the risk assessment it is mandatory to *define exactly about risks a module should be notified about.* A standard configuration taken from this work will simplify the integration of the risk assessment into arbitrary Grid systems. However, configuration possibilities should also be available in order to realise a customer-specific solution.

5.2.6 Decision and Risk Treatment

Based on the received risk factors the risk-enabled Grid modules decide whether and which action should be performed. During the *decision* process and before initiating a Risk Management action further risk assessments may be necessary in order to compare the advantages and disadvantages of a Risk Management activity. For example this applies to in the scheduling process:

Example 5.2.1
Let the probability of a resource outage of node x be too high with respect to the requirements of the job y executing on it. In the scope of a risk aware rescheduling, the PoFs for running job y on other resources are required in order to select the optimal resource.

In the *risk treatment* the execution of the selected Risk Management action is performed. Since the Grid Risk Management process is automated, the set of possible countermeasures is defined according to the features of the RMS. Only activities belonging to the set of supported features can be selected as countermeasures in the risk description and are considered as suitable risk treatments in the decision making. Consequently, *the steps for the decision making as well as the risk treatment for several Grid services have to be specified* to obtain a standard risk aware Grid solution. This will be adjustable in the configuration phase for the individual system of a Grid provider.

5.2.7 Residual Risk Reporting

The next step in the FERMA standard is *residual risk reporting*. This step is necessary since actions often cannot reduce the risk to zero and a residual risk still exists. In the Grid Risk Management this will be realised by monitoring the whole system and job states. Thus, no additional task has to be performed for reporting and observing residual risks. The monitored data has to contain information about all defined risk sources which are considered in the risk assessment methods. This data *has to be aggregated and formatted* in a uniform style, so that the risk assessment does not have to reformulate it and the data preparation is performed in the monitoring system. *Building statistics* from the data is mandatory in order to simplify the risk assessment and enable adaptive modifications. In this context, *evaluating the assessed probabilities and the real problems that have occurred* is very important in order to identify dynamic changes and to justify thresholds and models used in the risk assessment. Hence the modification process is very important to support a system specific risk assessment. Note that the evaluation process also includes a consideration of the ratio of *fulfilled and violated SLAs* by taking into account the guarantees, PoFs, revenues, and penalty fees. Since modifications of the methods used in the risk assessment have to be performed manually, controlling and adjusting the accuracy of the risk assessment also requires an interaction with an expert. Hence, it is not contained within the Grid Risk Management process and runs in parallel to it. However, the monitoring system should collect information to support such control activities.

5.2.8 Summary

To define a Risk Management process for the Grid, it is reasonable to use and adjust a standard instead of defining the process without any relationship to a standardised workflow. The FERMA standard was chosen as the basis for the Grid Risk Management since the standard is very detailed and most steps of FERMA – such as definition of objectives, risk reporting, decision making, and risk treatment – can be integrated in the Grid Risk Management process without any alterations. However, adjusting the standard was essential since the Grid Risk Management process should be automated and able to run without any human interaction.

In order to realise an automated Risk Management process, the risk identification phase is performed during the configuration of the Grid Risk Management process instead of integrated as a part of the risk analysis. Since FERMA is a standard, it supports the consideration of many functionalities and aspects which have to be re-focused in order to apply them in the Grid. The associated activities in the Grid Risk Management process are more targeted due to the main

and fixed defined objective to support SLA provisioning. Consequently, the tasks assigned to the risk description and estimation are reduced and the responsibility of the monitoring is re-defined to collecting information about compute nodes' hardware and software characteristics. The observation of the Risk Management process, which is performed in the monitoring step in the FERMA definition, is decoupled from the Grid Risk Management process since human interaction is required to re-define thresholds or risk assessment methods.

5.3 Targeted Risk Management – Using Risk as Decisive Factor

The Grid Risk Management process is defined to handle risky events or states such as an increased probability of a resource failure which can be identified by analysing collected monitoring data. The analysis is performed by the monitoring system installed in the Grid fabric itself and it notifies the risk assessment module after a warning or critical state is identified. This process does not exploit the full potential of risk awareness since various scenarios exist in which PoF information is beneficial to use [Djem 06, Voss 06]. The PoF estimations are supporting decision processes in the provider's RMS during the SLA negotiation and in the post-negotiation phase. In addition to improving the service provisioning, publishing risk/PoF for an SLA, increases the trustworthiness of providers [Voss 07c]. Envisioning the PoF as an additional component of an SLA, more end-users might be attracted to use the Grid even for business-critical computations. In order to enable a provider to publish failure probabilities within an SLA offer or to accept an SLA request, which defines a maximum acceptable PoF, risk awareness needs to be integrated in their internal processes – during the SLA negotiation and after committing an SLA. Taking into account risks in the provider's decisions involved in SLA provisioning is decoupled from the general Grid Risk Management process, which is initiated as a consequence of a risky event. However, the consideration of risk in the decision processes counts also among Risk Management, denoted as *targeted Risk Management* . The Grid Risk Management process developed has to be modified, since the targeted Risk Management process is not initiated because a risky event occurred. Further the communication between the risk assessment and the module responsible for the decision making and which has invoked the probability computation, has to be adjusted since the monitoring system is not the initiator of the Risk Management process.

The beginning of the targeted Risk Management process (depicted in Figure 5.2) equals the Grid Risk Management process since the definition of strategic objectives is followed by the risk identification. The next step is also denoted as monitoring, however, its meaning is adjusted: the monitoring is performed in the Grid module which will invoke the risk assessment due to use the estimated PoF value as a key aspect in the decision making. Consequently, the task monitoring differs from its original meaning in a RMS, which describes collecting data about characteristics of compute nodes or jobs. In the targeted Risk Management the monitoring process observes the occurrence of events, whose treatment decisively depend on a probability/ risk information. Following, the monitoring in the targeted Risk Management serves as an event trigger for the risk assessment. Resulting from this new meaning of monitoring, the periodical execution of the risk identification is dropped. Note that, if in addition to the

monitoring of the targeted Risk Management process, a classical monitoring system is utilised, the data collected therein can be used to periodically repeat the risk identification phase.

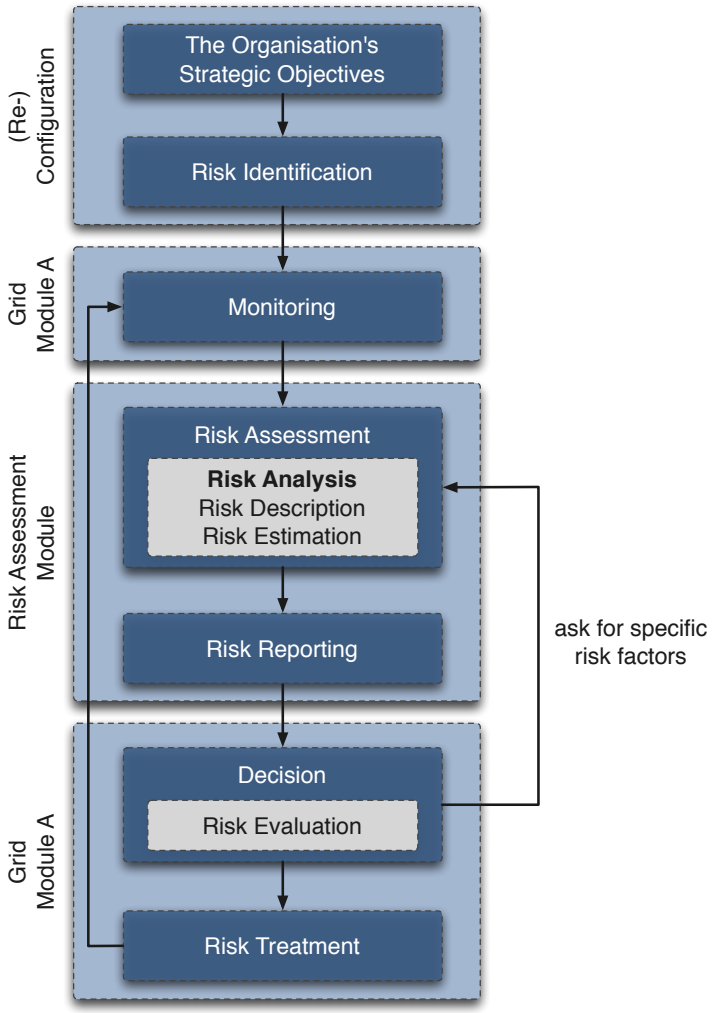


Figure 5.2: Targeted Grid Risk Management process when Using Risk as Decisive Factor

The targeted Grid Risk Management process further differs from the general one concerning the risk analysis. The risk evaluation phase is no longer part of the risk assessment since the risk evaluation contains the comparison of different risks and filtering of negligible risks. If a targeted Risk Management process is executed, a Grid module A has observed an event E, whose treatment should use a failure probability as a decisive factor. As a consequence, a comparison of different risks as well as a filter process must not be performed in the risk assessment since the requested PoF has to be used during the decision making. If the risk assessment would evaluate the estimated PoF as negligible and no reply is sent to the Grid module A, the handling of event E could not be resumed. Resulting, the targeted Risk Management process has positioned the risk evaluation as part of the decision making in the Grid module.

The risk reporting step is simplified in the targeted Risk Management process since the risk

assessment has to send the reply only to the Grid module initiated the PoF calculation. In contrast to the general Risk Management process, the risk reporting here does not send any information to the monitoring. As in the general Grid Risk Management process it could be necessary to ask during the decision making the risk assessment for additional estimations. The risk treatment is still separate from the decision-making and is the final state of performing a targeted Risk Management. After the risk is treated the Grid module resumes to wait on new events in the monitoring state.

The usage of risk in the context of SLA provisioning is beneficial to ensure an SLA fulfilment and also to gain users' trust. Using risk as a decisive factor in the RMS leads to the development of the targeted Risk Management, which has few modifications in comparison of the general Grid Risk Management process (see Section 5.2). The main difference is the task and responsibility of the monitoring step, which consists of the observation of events in the Grid module in the targeted Risk Management process. Furthermore, the risk evaluation is part of the decision making since the comparison of different risks and filtering of negligible risks must not be performed during the risk analysis.

5.4 Provider's Risk Management

The previous section presented the general and targeted Grid Risk Management processes. Thereby some aspects were identified which have to be defined before the Risk Management processes are integrated into RMSs. The necessity of a Grid Risk Management process was substantiated by the requirement of an automated execution without any manual interaction. A Grid Risk Management process needs manual input for the configuration only, all other processes execute without interaction. As illustrated in Chapter 4 Grid resource providers pursue the same objectives. To ensure an easy integration to arbitrary Grid technologies, most problems and questions identified for Grid Risk Management processes should be predefined. Since this thesis focuses on Risk Management of Grid resource providers, Section 5.4.1 – 5.4.8 investigate the following points from provider's point of view:

- A detailed specification of strategic objectives (including definition of policies) is required to set into relation the risks and countermeasures.
- The risk identification is initially done by an expert. It will be periodically updated for adjusting defined thresholds used in the risk assessment.
- A precise input specification of different risk factors will be necessary which are considered in the risk assessment model. This includes a definition of required and optional input, which influence a risk factor.
- Define policies to enable the provider to compare different risks and to decide whether a risk is negligible (either during risk assessment or for the decision making).
- After filtering risk information, the assessed values have to be published from the risk assessment module. A definition which Grid module is notified about which information is necessary.
- The steps for the decision making as well as the risk treatment for several Grid modules have to be specified.

- The monitoring data has to be aggregated and formatted. Methods have to be defined for the aggregation.
- Review risk includes comparing the assessed risks with occurred problems as well as how many and which SLA have been fulfilled.

Remind to differentiate between two Risk Management processes within a Grid. The risk/PoF assessment can either be initiated by the scheduler to use the estimated probabilities of failure as decision support or it can be estimated due to surrender values of a monitored event, e. g. a critical value of a hardware component has been recorded. The first case implies that the risk assessor adopts just the function for the estimation of the probability, whereas the risk evaluation is realised during the decision-making within the responsible Grid module. The second case implies that the risk assessor additionally performs the evaluation, compares risks, and filters negligible risks.

5.4.1 Specification of Strategic Objectives

When specifying the strategic objectives based on the requirement analysis (see Chapter 4), the fact has to be considered that Grid resource providers are commercial acting companies which targets to make as much profit as possible. To achieve this goal, providers want to accept as many SLA bound jobs as possible. However, the provider must only accept so many SLAs that the sum of penalty payments is as low as possible. The key aspect to balance both targets is the risk management integrated in the RMS since then for each SLA request the probability of failure is assessed and based on this value the decision is made and the revenue is defined. Minimising the number of SLA violations is not only important when considering penalty payments. An accurate failure ratio enables that the provider proves itself as a reliable contracting party. This is crucial since a good reputation of a provider is an additional decisive factor of consumers when selecting among different resource providers.

In the case that not all SLA bound jobs can be fulfilled, a policy can determine which jobs should be executed (see Chapter 8). Furthermore, policies can be used to define the minimum accepted profit margin. Such a limitation is important in scope of SLA negotiations in order to determine whether to accept or reject an SLA request. In particular the profit margin has to be considered when planning and performing risk reduction activities (see Section 7.4). To control SLA negotiations, policies have to determine whether requested PoF values should be accepted which could not be fulfilled. This means that internally the provider accept a higher PoF than the upper bound requested by the consumer. Policies may define the acceptable ratio between requested and estimated PoF. Note that in the Grid independent institutions or organisations can be implemented which statistically compare the ratio of published failure probabilities and failure rates. In context of the AssessGrid project [AssessGr 08], the Grid broker is responsible for checking the reliability of providers. If a provider is marked as unreliable, it can adjust the PoF and make this information available for end-users. Consequently, providers should be careful to allow higher internal PoF rates. In addition the EC-funded project GridEcon [GridEcon 08] also considers to develop reputation centres in order to evaluate the reliability and performance of Grid providers in scope of SLA provisioning.

5.4.2 Risk Identification

Risk identification considers threats/problems, their origin, and their consequences. Section 3.1.3 presented different approaches for performing a risk identification: Source Analysis or Problem Analysis, which can be used in scope of Objective-Based Risk Identification, Scenario Based Risk Identification, Taxonomy-Based Risk Identification or Common Risk-Checking.

From the perspective of a commercial Grid provider, the worst threats are SLA violations because of the penalty. Threats in scope of SLA provisioning can be described by processes, events, and reactions performed or occurring in the Grid fabric. Consequently, a scenario based risk identification is suitable for specifying events posing a threat of fulfilling all guarantees of SLAs. The cause of an unsuccessful job execution is a result from either the unavailability of sufficient resources on time or on problems within the job execution itself. In the computational Grid the unavailability of resources primarily conforms to the unavailability of compute nodes; for different Grid types another main resource type can be defined. The primary origins of events leading to an SLA violation in computational Grids are instabilities of compute resources; consequently, performing a source analysis is sensible. The unavailability or instability of resources is then again the consequence of another origin. If the resource outage has not been caused by a software problem (error in a software component or incompatible versions of interacting software), in most cases a hardware defect or instability is the origin. Intern sources of a resource unavailability are software problems or breakdowns of a system component, cooling system etc. Figure 5.3 depicts an example listing of various origins which can lead to a resource outage. In addition to internal sources, the risk assessment usually also considers external sources. In the Grid context external origins are, for example, the breakdown or outage of external Grid components, for which other providers are responsible, but also force majeure or natural disasters are counted among these.

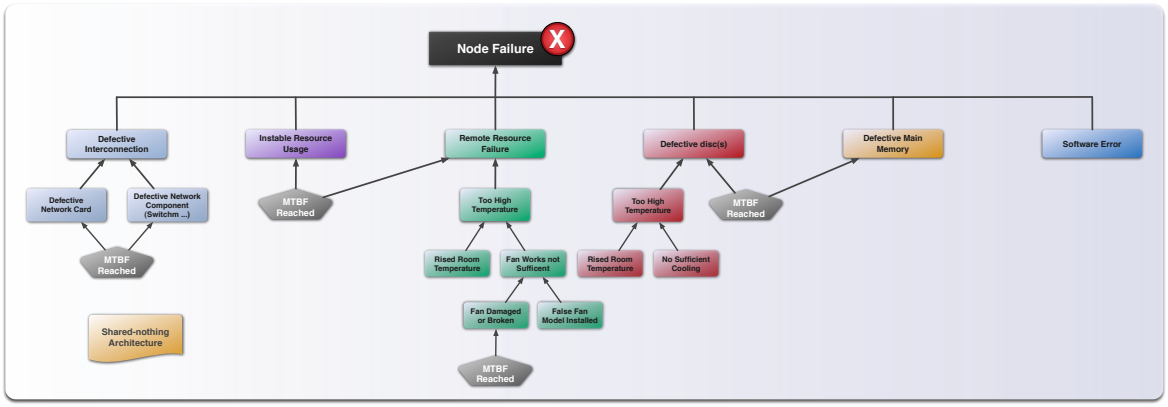


Figure 5.3: Origins and Problems for a Resource Failure

Figure 5.4 uses a tree structure in order to point out the chain of origins which can be responsible for a hardware defect or instability. It is important to remark that a problem might occur and no specific origin can be identified. The tree structure only shows conceivable origins and should not be assumed to be a complete catalogue. The origins listed have been identified in a shared-nothing architecture [Cull 99] in order to leave out of consideration

competing accesses or mistakes caused by other compute nodes. If either RAM or disk space is shared, the set of origins is significant larger.

SLA bound jobs not exclusively using compute nodes of a cluster, might fail because of an outage of a used system. Consequently, in addition to the problems which might lead to an resource failure, other origins can lead to an unsuccessful job execution. Figure 5.4 depicts several exemplarily origins, which may lead to a failed job execution. The Grid service provider is however not responsible for all listed aspects. The service consumer has to ensure that the job description is correct and all requirements are defined by service and guarantee terms in the SLA. For example, if a necessary library for the job execution is not specified in the SLA, this will cause a job abort if this library is not *coincidental* installed on the compute node. Such errors are not in the responsibility of the Grid provider since a correct and complete description of the job requirements is an obligation of the service consumer. Consequently from the provider's perspective such scenarios need not be considered in the risk identification.

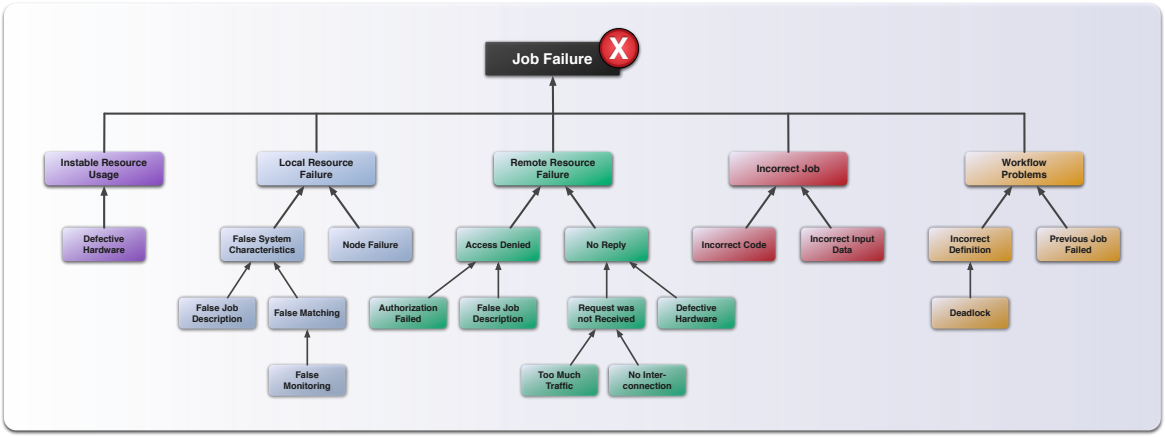


Figure 5.4: Origins and Problems for a Job Failure

The administrator can determine based on the typically involved resources in the execution of SLA bound jobs, which problems occur and can guess initially the frequency of occurrence. By collecting, aggregating, and evaluating monitoring information these values can be automatically adjusted. Thus, the expert's definition is validated or corrected and thresholds are dynamically adjusted. A dynamic adjustment is important since, for example, the frequency of resource outages increases with the age of the hardware since hardware defects occur more often as in brand new hardware

5.4.3 Input Specification of Different Risk Factors

Optional and required input parameter for the risk assessment has to be defined based on the applied risk model and estimation process. Since the risk identification has to be performed previous to this definition, a classification of input parameters cannot be done in general at this stage. Note that if defining parameters, the set of the required input data should be as small as possible. Based on the required data, the risk assessment computes an *initial PoF*.

By taking into account more information, the risk assessment is able to state the PoFs more precisely. The more information is available and considered within the PoF calculation, the more accurate are the PoFs.

Data collected by arbitrary monitoring systems in the Grid fabric significantly vary in type and frequency. Due to simplify an integration of risk assessment methods developed for one provider in a different RMS of another provider, as few as possible restrictions should be made. This is of particular importance in order to support the interaction of risk assessment with different software solutions which have been established in the cluster operation of a provider. Consequently, to balance accuracy and reusability of the risk assessment methods developed, it is desired to have a small set of required input parameter. Chapter 6 presents the initial risk assessment model whose considered data conforms to the required input. Chapter 7 extend the PoF estimation by taking into account the initiation of FT-mechanisms during the SLA negotiation. These enhancements are optional parameters. Chapter 9 the basic scenario used for evaluating the benefit of applying Risk Management.

5.4.4 Policies for Comparing Risks and Definition of Negligible Risks

Since the focus of the integrated Risk Management process is on SLA provisioning, the risk assessment is primarily initiated from the RMS to take into account PoFs during the scheduling. In this case the risk assessment module (also denoted as *risk assessor*) will not filter any PoFs and publishes all estimated values. As a consequence, the risk evaluation is performed within the Grid modules and does not count to the responsibilities of the risk assessor. Chapter 7 and Chapter 8 present example policies used for comparing different risks in the scheduling. Furthermore, the risk estimated during the SLA negotiation decisively influences whether to provider accept or rejects an SLA offer by taking into account revenue and penalty. Hence, in this field of application negligible risks do not exist in the original meaning which classifies a risk as negligible if it is too low to initiate any countermeasure.

In particular, the initial risk is determined during the SLA negotiation and, because of threats related to resource availability, it may be modified after agreement. In the general Grid Risk Management process managing a risk modification is performed more frequent than in scope of the targeted Grid Risk Management process, since a monitored event leads to a change of the PoF estimated during the SLA negotiation. The risk assessor decides dependent on the modification whether to classify the change as negligible and refrain to publish the new PoF. During the configuration, negligible PoF variations have to be defined by the use of thresholds. In addition the definition of an upper bound for the maximum negligible risk is reasonable.

5.4.5 Notification of Grid Modules

The publication of risk information is realised by sending appropriate messages to a preselected set of Grid modules. It is necessary to configure, which modules of the Grid provider should be notified. In the targeted Grid Risk Management process the notification is in default limited to the module which has sent the request. Any arbitrary software component may send a request as long as it is assigned to the same administrative domain and it uses the estimated risk or PoF as a criterion in its decision-making. Consequently, in general no limitations exist

concerning the software modules initiating a risk estimation or the number or kind of modules notified about estimated values. In most cases also in the general Grid Risk Management process only one Grid module is responsible for the decision-making and risk treatment and consequently, risk information is usually only send to one Grid module. Dependent on the scope of operation of the risk assessment, following modules are conceivable to be notified:

- Scheduler
- Fault-tolerance manager
- Negotiation manager
- Security manager
- Monitoring GUI

The primary Grid module of the RMS to be notified is the scheduler, which processes risk information and is integrated to perform Risk Management processes. Its tight relation to the risk assessment is caused by the responsibility of the scheduler for delegating the job execution since this contains generating the matching of resources and jobs as well as timing the job execution. If the initiation of FT-mechanisms does not belong to the responsibility of the scheduler, the responsible component, denoted as fault-tolerance manager, should be notified about PoFs estimated. Finally, in order to benefit of FT-mechanisms in the Risk Management, this module should consider PoF in its decisions.

Risk information can be also important for the module of the provider negotiating SLAs with contractors under the terms of WS-Agreement [Andr 07] or WS-Agreement Negotiation [Andr 06]. It can be used to make a preselection of possibly acceptable SLAs or determine an SLA rejection without any interaction with the scheduler. Such a behaviour is imaginable, for example, for workflow jobs. The negotiation manager can determine based on the job description the dependencies of sub-jobs and the total computation time. Based on this information the risk assessment is able to estimate a minimum PoF. If this minimum value is higher than the maximum bound of the contractor, the negotiation manager can reject the SLA request without initiating the evaluation in the scheduler which would make an advance reservation according to service and guarantee terms, i. e. it would check the SLA's feasibility.

Notifying security managers is meaningful if the risk assessment considers security aspects and can identify related threats. For example, a risk assessment could determine the probability of a *Denial-of-Service* (DoS) attack based on monitoring information about SLA negotiations or job activities. A DoS in scope of SLA negotiation can be identified if many tentative SLA requests are sent from the same consumer. Spamming with not serious SLA requests leads to many feasibility checks in the RMS and keep providers from processing serious SLA requests. Monitoring job activities can point out a DoS attack, if several jobs send requests to specific components of the provider's infrastructure. By using authentication mechanisms, those job would probably have no access to arbitrary components, however the load produced by these activities thwarts the system and network.

In the case administrators use a GUI for observing hardware components in the infrastructure, they see characteristics collected by monitoring systems. If the data visualised is also used by the risk assessment in order to determine the PoF of a compute node, presenting the PoF in the GUI would be beneficial. Since PoFs will dynamically change based on the hardware

information, an appropriate mechanism has to be used to make the GUI applicable and system efficient. For example the GUI can provide the user with PoF information on demand: if an administrator is interested in the PoF estimation, a request is send to the risk assessment. This scenario shows the implementation of a targeted Grid Risk Management process, in which the GUI is the initiator of the risk assessment in order to show this information to the administrator. At this point, the Grid Risk Management process passes into the classical Risk Management since only the PoF estimation is automated generated and the decision-making and risk treatment is performed by a human being – the expert.

5.4.6 Decision Making and Risk Treatment

Developing processes for the decision making is in focus of this work. They base on the capabilities and features supported by the RMS, which are applicable in the risk treatment. By considering features only as possible options, the developed decision processes can even select the best Risk Management activity if not all features are supported. Consequently, the mechanisms developed are applicable in arbitrary planning based RMS. The risk treatment conforms to the initiation and execution of appropriate and selected functions according to the decision made.

The decision-making can either locally concern one single job or can be performed globally by considering all jobs in the system. Chapter 7 and Chapter 8 present workflows and criteria for the decision-making. It is important to remark, that usually the decision made for one job affects all other jobs in the system. These consequences are based on the fact that by assigning to a job new or more resources, the number of alternatively useable resources for all other jobs on the cluster is reduced. This number is however important for all SLA bound jobs since free resources are crucial to compensate for resource failures. The underlying risk assessment model decides whether these threats result in modified PoFs. A modification of PoFs for any job leads to a re-evaluation of its risk in order to decide whether to treat it or mark them as negligible.

5.4.7 Aggregation of Monitoring Information

The aggregation of monitoring information depends on the input data of the risk assessment, since only data has to be collected, analysed and aggregated, which is used in the PoF estimation. A standard approach for the data preparation in data mining is clustering which groups homogeneous information under the consideration of specific tolerance intervals and estimates the groups' mean values. [Birk 07] describes the mechanism of clustering applied to aggregate monitoring data of CPU temperatures. The core idea of performing aggregation is to not assign data to fixed superior thresholds, rather to dynamically estimate the average value of a group based on the monitoring data.

Let $T = (t_1, t_2, \ldots, t_n) = (t_i)_{i=1}^{n}$ be a time series that was obtained from monitoring a computing node. Let $a_1 < a_2 < \ldots < a_m$ be threshold values defining the classes of the observations with respect to the data measured. The set of values a_j, $j = 1, \ldots, m$ have to be estimated by experts, to reliably determine which parts of the scale is reactive to small changes. For example, an expert can observe that CPU temperature between 30°C and 35°C does not

make significant impact on the stability of the processor, but above 35°C a small variation in temperature can (but not necessarily does) cause problems in the smooth operation of the device.

Under this circumstances, time series T can be compressed by gathering adjacent observations that do not deviate more than a predetermined threshold. Mathematically formulated:

$$\langle T \rangle \quad = \quad (\vartheta_1, \ell_1; \vartheta_2, \ell_2; \ldots; \vartheta_N, \ell_N) = (\vartheta_k, \ell_k)_{k=1}^{N}, \tag{5.1}$$

where for all $i = 1, \ldots, n$ the following relationships hold:

$$t_i \in [a_p, a_{p+1}), i = L_r + 1, \ldots, L_r + \ell_{r+1} \text{ with } L_r = \sum_{q=1}^{r} \ell_q \tag{5.2}$$

for some $p \in \{1, \ldots, m-1\}$ and $r \in \{0, \ldots, N-1\}$, and

$$\vartheta_k \quad = \quad \frac{1}{\ell_{r+1}} \sum_{i=L_r+1}^{L_r+\ell_{r+1}} t_i \tag{5.3}$$

for all $k = 1, \ldots, N$. These relationships mathematically formulate that for any $r \in \{0, \ldots, N-1\}$ all elements t_i in time interval $i = L_r + 1, \ldots, L_r + \ell_{r+1}$ of the time series belong to the same class, i. e. they are all included and thus represented by interval $[a_p, a_{p+1})$ for some $p \in \{1, \ldots, m-1\}$. In particular, for $k = 1, \ldots, N$, ℓ_k and $L_{k-1}+1$ denote the *length* and the *index of the first element* of the k-th block in T with elements belonging to the same class, respectively.

In this particular compression the *consecutive* values t_i of the time series have been substituted which belong to the same class with their *arithmetic average*, and also store the *number* of values t_i that are substituted. Thus, for all $k = 1, \ldots, N$ two pieces of information are stored in $\langle T \rangle$ for each block of data of length ℓ_k in T. The quantity $\rho = 2N/n$ is denoted as *rate of compression*. It is clear that the wider the intervals $[a_p, a_{p+1})$, $p = 1, \ldots, m-1$ (or equivalently, the lower index m is set) are, the better is the rate of compression. Furthermore, there is also a clear trade-off between rate of compression and amount of information *preserved* in $\langle T \rangle$ as compared to T. In the following, the compression method is illustrated by a simple example.

Example 5.4.1
Let a sensor collect data of CPU temperatures of a computing node. 5-minute time steps are used and monitoring data was collected for 1 hour. Resulting is the length of the time series $n = 12$. Further the following $m = 3$ threshold values have been defined from experts: $a_1 = 30$°C, $a_2 = 35$°C and $a_3 = 38$°C. The sequential data (in °C) which was collected is:

$$T = (32, 33, 31, 34, 35, 37, 36, 33, 32, 33, 34, 33). \tag{5.4}$$

Using notations and information presented above, it can be easily verified that in this case $N = 3$ with $\ell_1 = 4, \quad \ell_2 = 3, \quad \ell_3 = 5$, and $L_0 = 0, \quad L_1 = \ell_1 = 4, \quad L_2 = \ell_1 + \ell_2 = 7$. Thus,

$$t_i \in [a_1, a_2), \quad i = L_0 + 1, \ldots, L_0 + \ell_1, \tag{5.5}$$
$$t_i \in [a_2, a_3), \quad i = L_1 + 1, \ldots, L_1 + \ell_2, \tag{5.6}$$
$$t_i \in [a_1, a_2), \quad i = L_2 + 1, \ldots, L_2 + \ell_3. \tag{5.7}$$

Hence, carry out the following compression:

$$\langle T \rangle = (32.5^{\circ}\mathrm{C}, 4; 36.0^{\circ}\mathrm{C}, 3; 33.0^{\circ}\mathrm{C}, 5). \tag{5.8}$$

In particular, the rate of compression is $\rho = 6/12 = 1/2$.

Similar methods can be used for aggregating arbitrary monitoring information. At this stage no detailed specification is possible since the aggregation depends on the input data of the risk assessment.

5.4.8 Risk Review

If adaptive methods are used for the risk assessment, the risk review process is crucial. Reviewing the risk means to compare PoFs estimated with the actually occurrence of the corresponding events. For example, the risk assessment determines the PoF for a resource within a time-frame. If then the resource stability is monitored during this time-frame and no resource failure occurred, this information should be used to improve the risk assessment. Note that a critical mass of information has to be collected before adjusting methods or thresholds, i. e. a single observation must not effect the risk assessment. Risk reviewing also ensures that the risk assessment is automated and dynamically adjusted to changes in the system. In this scope a further example for a risk review is to prepare statistics about the fulfilment of SLAs. By categorising SLAs according to their service or guarantee terms and relating these to the number of SLA violations, correlations can be maybe identified which are not reflected in the risk assessment. For example higher or lower failure rates can exist for SLAs with specific requirements or c be constituted because of timing differences of their execution – day or night/week or weekend.

In the Grid, the expected loss is a priori known value – the agreed penalty fee – and consequently the risk reviewing has not to evaluate the accuracy of the risk and can focus on probabilities. The risk review is tightly coupled with the methods applied in the risk assessment. Its results serve for the validation of the estimated PoFs and forms the base for modifying and adjusting the assessment model, such as adjusting the quantifier of different thresholds or considering additional input data. The first step of the risk review process can be executed independently from the risk assessment model in the RMS's monitoring system: various statistics about the SLA fulfilment and resource outages can be generated in order to use data mining methods for identifying correlations after sufficient data has been collected.

5.5 Summary

This chapter described two different Grid Risk Management processes which are derived from the FERMA standard. The differentiation of a general Grid Risk Management process and a targeted Grid Risk Management process is necessary since in the first case a risky event occurred, which has to be treated if it is not negligible. In the targeted Grid Risk Management process risk/PoF influences decisions in scope of SLA provisioning. Such an integration of risk awareness exploit the full potential of the PoF calculation for resource failures and SLA

violations. To configure and implement both Grid Risk Management processes, some questions have to be clarified as pointed out in Section 5.2. Section 5.4 described whether these can be defined in general for Grid providers or in which scope they have to be defined. Many aspects depend on the risk assessment model and can be specified after its definition. As a consequence, the next chapter details the underlying model of the Grid provider as well as the risk assessment.

Chapter 6

Risk Assessment and Underlying Model

A precise definition of the risk assessment is important in order to define Risk Management strategies. This chapter presents the initial PoF calculation which might be modified if risk reduction is applied. Risk reduction can be applied to lower a PoF since this is not acceptable. In Chapter 7 the PoF modifications are presented if a specific risk reduction technique is used. Consequently, the enhancements of the PoF calculation are optional whereas the model presented in this chapter reflects the basic idea and the required input data. Since the risk assessment model was developed based on a specific Grid fabric model, this has to be precisely described first in Section 6.1. After presenting the risk assessment model in Section 6.2, the basis for the defining Grid Risk Management processes is formed. Section 6.3 gives an outlook how risk awareness is integrated in the Grid.

6.1 Underlying Model of the Grid Fabric

To precise the Risk Management processes in the Grid fabric, the definition of an underlying model is necessary because of several reasons: First, the job definition and job handling is essential for SLA provisioning and therewith for the risk assessment and decision making. Secondly, the underlying RMS concept strongly influences the processes in the Grid which can be enhanced by risk awareness – using the targeted Risk Management. Thirdly, it influences the capabilities of risk treatment, i. e. *which actions can be performed to manage risks?*

Section 6.1.1 defines basic assumptions made concerning the Grid fabric. Section 6.1.2 describes the underlying model concerning job definition and job execution which is affiliated with SLA provisioning. Afterwards, Section 6.1.3 presents details about the underlying RMS and job processing. Essential in the context of risk treatment is the usage of *Fault-Tolerance* (FT)-mechanisms which are summarised in Section 6.1.4.

6.1.1 Grid and Grid Fabric

Due to its focus on supporting SLA provisioning of the Grid provider, the Grid Risk Management process has a limited field of application in contrast to the FERMA standard which can be used for Risk Management processes of all activities in various organisations. The limited field of application enables an automated execution of the Grid Risk Management process since features for the risk treatment can be generally defined a priori and during the

configuration phase these can be adjusted according to the provider's specific environment. Obviously, implementing Grid Risk Management processes relies on the environment in which Risk Management should be integrated. Hence, to define in the development of a Grid Risk Management process the risk treatment and the decision-making, a model has to be specify in which system the Risk Management should be integrated and whether the underlying architecture influences the risk treatment. This section details the key aspects of the system in order to clarify the assumptions made in the development of the Grid Risk Management process.

Resource providers buy often homogenous clusters, since the homogeneity reduces the configuration and maintenance effort for operating the system and vendors offer higher discounts when selling the same hardware in great quantities. Consequently, the computational Grid prevailing consists of homogenous clusters, as observable in the hardware listings of the Grid'5000 [Grid5000 08]. Note that the differences between clusters operated by the same or by different resource providers can be large. The heterogeneity of clusters operated by the same provider results from several reasons: all clusters are not contemporaneously bought and, since a provider will always buy up to date hardware, the hardware of different clusters vary concerning their processor speeds, storage capacities, etc. Furthermore jobs might ask for a specific architecture to be executed on or a powerful graphic card. In order to run more jobs, a provider might buy a cluster fulfilling specific hardware requirements.

Each cluster is managed by a RMS and a superior module is responsible for forwarding SLA request to the RMS of a cluster, denoted *gateway* (cf. Figure 6.1). An experimental comparison of the EGEE Grid and the Grid'5000 have shown that a production Grid such as EGEE [EGEE 08] uses the Internet for communication between different Grid sites, whereas the Grid'5000 clusters communicate via Gigabyte Ethernet links [Glat 06]. Hence, the communication between clusters of the same provider will probably be significant faster than the communication to another Grid site. To simplify the model, this thesis considers the operation of one cluster. This limitation has no effects on the development of Grid Risk Management processes since the Grid Risk Management process is integrated in the RMS and a RMS only manages one cluster. If in a Risk Management process a differentiation of clusters operated by another or same provider is necessary, the aspects to be considered are pointed out. Since a RMS is responsible for the job execution on one single compute cluster which are usually homogeneous, it manages n homogenous compute nodes similarly installed and configured. Due to the nodes' homogeneity, speed constraints in an SLA are fulfilled to the same extent by each compute node. Hence, a job can be executed on each node, i. e. the scheduling does not have to compare performance or configuration characteristics of different resources. If the provider operates several different clusters, the gateway decides on which cluster the job should be executed. Hence, if several clusters fulfill all constraints of an SLA bound job, the gateway forwards the request to the RMS managing the cluster with the slowest processor, etc. Following this strategy, the more powerful resources are kept free for later arriving job requests perhaps having stronger performance requirements.

In order to avoid unpredictable side effects, on the cluster only jobs can be executed which have been submitted through the RMS. Hence, users have to use the RMS as a single entry point. This limitation is necessary in order to assume that the RMS has full knowledge of the jobs running and scheduled on the cluster's nodes.

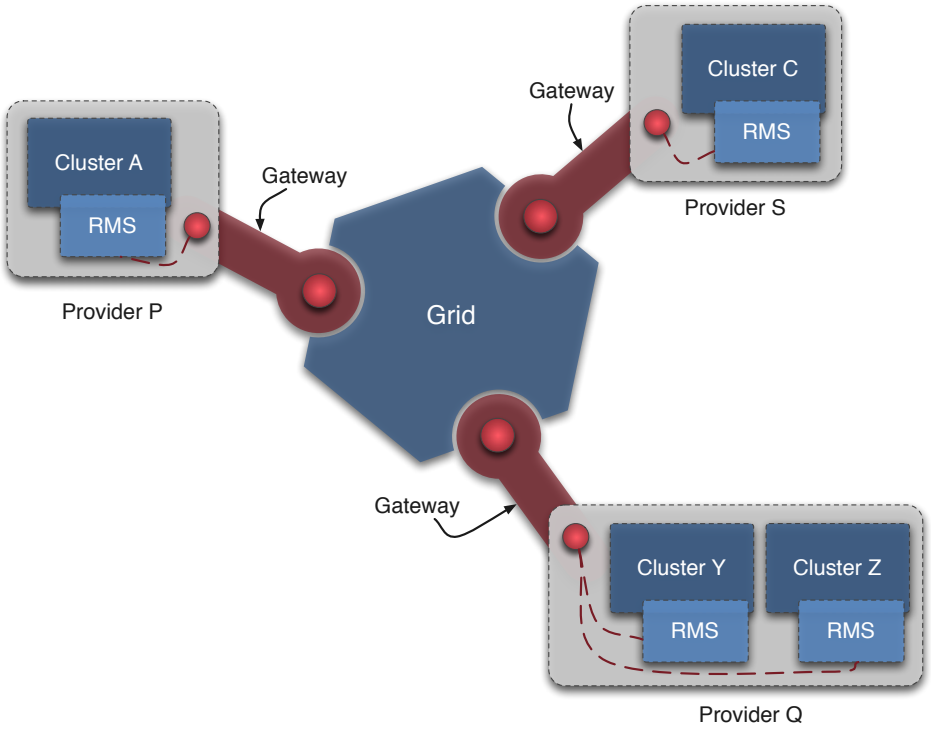


Figure 6.1: Assignments of Clusters and RMSs

To implement the automated Grid Risk Management process, a specification of the environment is required in order to define features for the risk treatment and develop decision-making processes. Summarising, the model considers that a RMS manages a homogeneous cluster and without imposing a restriction a provider only operates one cluster on which jobs can be executed.

6.1.2 Jobs

The environment of the Grid Risk Management process was defined in the previous section. The other key components addressing the SLA provisioning are the jobs executed on the provider's cluster. In scope of the decision-making and cluster operation, a differentiation of job types is necessary since jobs have different requirements and states. Requirements which can be defined in the SLA have to be detailed in order to point out key functionalities of the underlying RMS. Further, the risk assessment has to define the *Probability of Failure* (PoF) of resources and SLA bound jobs. To develop the risk assessment algorithm questions such as the following have to be answered: *What are the terms of an SLA which might cause a violation? What is the expected loss of an SLA violation?* The underlying model answering those questions is presented in this section.

SLAs may define criteria of the job execution. As a result, jobs are differentiated to be performed in best-effort, bound to a negotiated SLA, or under negotiation. After the provider receives an SLA bound job request, the RMS internally evaluates the risk of agreeing the

associated SLA by making a tentative resource reservation. Based on the risk aware resource selection, the provider can decide whether to agree or reject the SLA request.

An SLA may define arbitrary service and guarantee terms (see Section 3.3). If any of these terms is violated, the provider has to pay a defined penalty fee. To simplify the risk treatment in the automated Grid process, some assumptions have been made concerning the terms of an SLA bound job. The negotiation is realised by *WS-Agreement* (WS-AG) or *WS-Agreement Negotiation* (WSAN) – protocols defining the negotiation activities and states. These protocols, however, do not define the content of an SLA. To define the content, *Service Description Terms* (SDT) and guarantee terms are used. SDTs are described by using the *Job Submission Description Language* (JSDL) [Anjo 06] developed by the Open Grid Forum (OGF) [OGF 08]. SDTs can define the number of resources, speed constraints, the amount of memory etc. According to the requirements defined by SDTs with JSDL, the model has no limitations; the RMS must be capable to reserve the required hardware resources as demanded. Guarantee terms can be used to define various objectives. The RMS used for the Grid Risk Management process limits the supported guarantees since it has to support a validation of their fulfilment as well as adequate mechanisms to ensure these. The RMS enhanced with risk awareness has to support the key feature to define time constraints. Accordingly, an SLA bound job may have a latest finish-time, earliest start time, and duration (see Figure 6.2). If the earliest start time is not defined in the SLA, it equals the time of the negotiation initiation. If the earliest start time and latest-finish-time (deadline) are defined, the SLA has to define also the duration of the job. If no deadline is defined, the provider's time period for fulfilling the SLA is unlimited. Consequently, it is a best-effort job concerning the time and when running it the provider has 'only' to ensure that the required resources are generally available. If an SLA does not contain any guarantee term, the associated job is classified as a best-effort job.

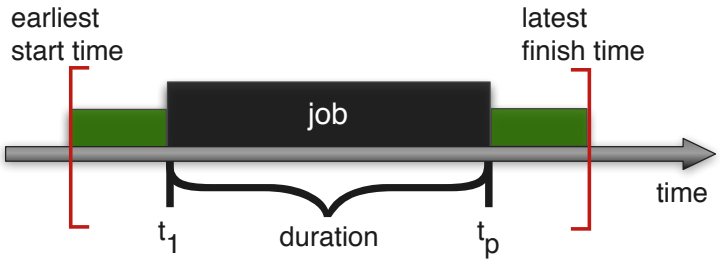


Figure 6.2: Job Slot Bounded by Earliest Start-time and Latest Finish-time

During the SLA negotiation, the provider estimates the PoF for the SLA based on several input factors such as the execution slot, the PoF of the resource reserved, as well as the pre-estimated availability of alternative resources. Since a resource outage is the most critical point for SLA provisioning, the probability of an SLA violation strongly depends on the probability of failure of the resources used. Using a more stable resource lowers the PoF for an SLA violation since in the case of a resource outage alternative resources have to be found on which the execution has to be resumed. Dependent on the hardware and configuration of compute nodes, differences between the stability of resources of the same cluster exist. Since a RMS manages only nodes of one cluster and those nodes are homogeneous, this work acts under the assumption that their stability is also very similar. Hence, is it likely that resources have the same probability of failure. However, the monitoring information and risk assessment

model can point out that PoF of resources vary. In this case it is crucial to differentiate which resources are reserved. In this model, similar behaviours of all compute nodes of a cluster are assumed as default and highlight a differentiation of resource stabilities if appropriate. Note that this equation is no limitation for the overall Risk Management idea, since the resource stability of compute nodes of different clusters may vary significant. Consequently, accurate estimations of the PoF when executing a job on a specific cluster is valuable.

The assigned execution slot of a job j is described by $s(j) = r_i|t_s - t_e$ where r_i are the resources used, t_s the planned start time, and t_e the planned end-time. Further, $r \in s(j)$ denotes that resource r is involved in the execution of j. The PoF for an SLA bound job j estimated during the reservation process is referred by $P_f(j)$ and equals the upper bound for an acceptable PoF. To accept more risk than offered, the provider could determine an internal accepted upper bound $P'_f(j) = mP_f(j)$ for an arbitrary factor m. This enhancement does not change the model.

To determine the risk of agreeing an SLA bound job, the PoF as well as the expected loss have to be taken into account. The PoF is computed during the SLA negotiation in the RMS. The expected loss can be defined in multiple ways: the loss, which will definitely occur and has therefore not to be estimated as an expected value, is the penalty fee, which has to be paid by the provider. On the other hand the contractor is disappointed when an SLA is violated and consequently, multiple SLA violation will endanger a good reputation of the provider. The effects of an SLA violation in scope of reliability and trustworthiness are hardly to estimate. Consequently, the model defines as the (expected) loss the penalty the provider has to pay in the case of an SLA violation. More precisely, the penalty is considered as a fixed value the provider has to pay if any of the guarantees defined in the SLA has been violated, for which it is responsible. This assumption simplifies the potentiality offered by the SLA construct. Generally, a penalty fee could be defined for each *Service Level Objective* (SLO) and the penalty definition may be described by a function or a fixed value. Consequently, some systems define the penalty based on the number of defect resources per unit time during the job execution or as a linear function over the time lapsed between job completion and its deadline [Yeo 05]. If the penalty would be defined to depend on the number of resource outages of the delay of the job completion, it is necessary to determine the expected penalty to estimate the risk. Considering a fixed value simplifies the evaluation of developments, but modifying the penalty definition does not influence the Grid Risk Management process and consequently no restriction follows from this simplification.

To define the Grid Risk Management process supporting the provisioning of SLAs, the key assumptions made for the jobs have to be specified in order to develop the decision-making. In the selection of a Risk Management activity, it has to be considered whether the job is bounded by an agreed SLA, is under negotiation, or will be executed in the context of a best-effort service. To determine PoFs, the risk assessment requires information about aspects leading to an SLA violation. The model assumes that the RMS supports in the scheduling the consideration of time constraints in order to allow the definition of earliest-start time, latest finish-time, and job duration in the SLA. Further, penalties are fixed values to be paid if any guarantee was not fulfilled as defined in the SLA. Hence, the risk equals the product of PoF and penalty.

6.1.3 Processing Jobs in the RMS

The targeted Grid Risk Management process (see Section 5.3) has to be interlaced with the processes in the RMS which are responsible for the SLA provisioning. Based on internal software components, RMS handles arriving SLA requests differently. The main workflows integrated in the preparation and management of the job execution have to be specified in the model to determine the potential starting points for integrating targeted Risk Management processes.

When an SLA request is received, the module responsible for the SLA negotiation forwards the request to the scheduler before deciding whether to agree or reject it. The scheduler is a planning based one in order to avoid additional uncertainties for SLA provisioning which exists in queuing based systems (see Section 1.2.3). The scheduler tries to make an advance reservation for the job which fulfills all requirements of the SLA. The advance reservation has to reserve as many resources as requested, each of the reserved ones has to fulfill the hardware and software requirements, and the reservation slot has to be valid concerning possibly defined time constraints. The negotiation module is notified from the scheduler whether or not the job execution is possible according to the SLA. It then decides about acceptance or rejection of the SLA request. This decision can be influenced by customer and market policies defined from the provider's management unit. Note that best-effort jobs are handled from the scheduler in the same manner, i. e. for those also an advance reservation is made and a RMS could reject such a job request. After planning a job execution during the SLA negotiation, the execution

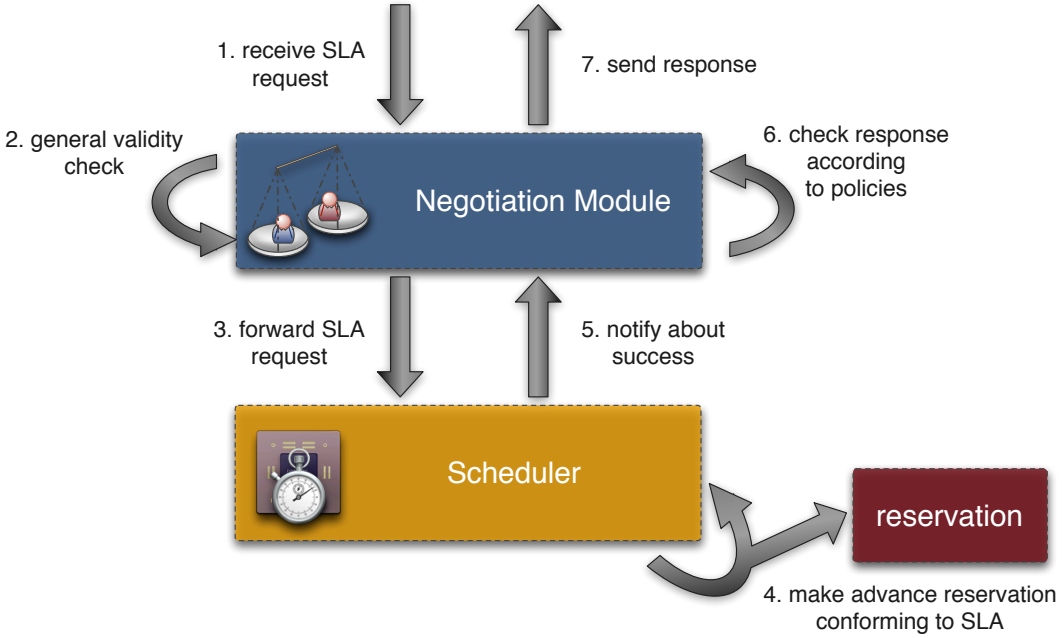


Figure 6.3: Workflow of Processing an SLA Request

slot may be modified as a consequence of resource outages and an initiated replanning of all jobs. If a resource has failed, the number of available resources is reduced by the failed number. Therefore the *Mean Time To Repair* (MTTR) will not be taken into account since a reboot and a hardware defect usually result in significantly different durations of resource

unavailability. A reboot usually takes only some minutes if the technical staff is working and monitoring the resource availability. To repair a hardware defect, the provider's policy can specify the expected time to repair, e. g. policies may define that all hardware defects are repaired immediately or only during general maintenance intervals. Even if hardware defects should be repaired immediately, technicians perhaps have to wait some hours (or days) for new hardware components. Consequently, a reboot cannot be put on the same level as a hardware defect. If no differentiation is made, the duration of a failure is relatively high, as the analysis of Grid'5000 data has shown in [Iosu 07]. Here, Iosup et al. have estimated an average duration of a failure – node's downtime – of 14 hours. According to their analysis this high value might also result from the fact that no technical staff is usually working on the weekend in the cluster centres of universities or research institutes. Consequently, a node's failure which can be solved by a reboot would not be repaired during the weekend. Since a fine grained information basis would be necessary to sensibly use downtimes in the scheduling, the model simplifies the scheduling by not considering periods between occurred failure and re-availability of a resource. After a failed resource is available again, the schedule will be replanned. If a resource is re-available after a few minutes, effected jobs can be resumed on-time on the re-available resource. As a consequence, not considering the MTTR is no restriction and each modification concerning the number of available resources initialises a replanning of the schedule.

To interlace targeted Grid Risk Management processes with the job processing in the RMS during the SLA negotiation, the workflow of making an advance reservation seems to be suitable. Combining the reservation making with PoF information enables the provider to publish failure probabilities or to accept upper bounds of PoFs defined by the contractor. Chapter 7 presents the targeted Grid Risk Management process for SLA bound jobs during their negotiation. The management of resource failures is realised through an initiated replanning of the schedule. Consequently, linking replanning processes with targeted Grid Risk Management processes integrates risk awareness in the RMS to higher the provider's profit. If an unstable resource has been detected from the monitoring system, the general Grid Risk Management process evaluates the initiation of appropriate FT-mechanisms instead of replanning the whole schedule. The decision-making processes in the post-negotiation phase are described in Chapter 8.

6.1.4 Fault-tolerance Mechanisms

In the computational Grid jobs are executed on several compute nodes in parallel and have often a long runtime of weeks or months, e. g. performing medical analysis or simulation. The SLA provisioning for such jobs is critical if these are also defining time constraints since a resource outage before the job completion probably result in not meeting the deadline. The reason is that in most cases the deadline is too close in order to complete the job until the deadline when restarting it from the beginning. Means to avoid a complete job restart are essential if an SLA violation should be prevented after a resource outage.

The most common and profitable mechanism to avoid a complete job restart is to make snapshots during the job execution and store these snapshots - also denoted as *checkpoints* [Ande 81] – somewhere in the network. Checkpointing can be used in order to resume

the job on the same resource after this is restarted. However, a hardware defect can lead to a long downtime, and accordingly checkpointing is often combined with the FT-mechanism migration. A job migration describes that the checkpointing data is transferred to another compute node and the job is resumed there from the latest checkpoint. In this scope it is important that the checkpointing data must not be stored on the resource used since after a resource outage this data could not be accessed. Self-organising FT-mechanisms based on checkpointing and migration have been already developed and can be used from Risk Management processes [Hove 06a, Wrze 05].

Generating checkpointing costs time. In particular, the checkpoint generation of parallel jobs having a long runtime, a high memory utilisation, and intensive communication with sub-jobs performed on other compute nodes, can need several minutes or even hours in which the job is paused and not computing [Ouel 05]. Dependent on the checkpointing frequency, the total time used for the checkpointing generation can lead to a significant lower time interval useable for the job computation. If a shorter runtime of the job is provided than requested within the SLA, the provider does not fulfil the SLA constraints. To ensure that sufficient time is available for the job execution in relation to the requested execution time within the SLA, the resource reservation time has to be extended in order to compensate for the time required for generating the checkpoint (see green slots in Figure 6.4). The runtime extension for enabling checkpointing demands for planning this FT-mechanisms in the initial reservation phase during the SLA negotiation. If checkpointing should be initiated after the contract conclusion, the duration has to be extended which may lead to a conflict in a planning based system, i. e. the duration cannot be extended because with the extension the reservation cannot be put in the schedule. Hence, planning checkpointing after the initial reservation phase is possible, however, performing it during the SLA negotiation should be preferred.

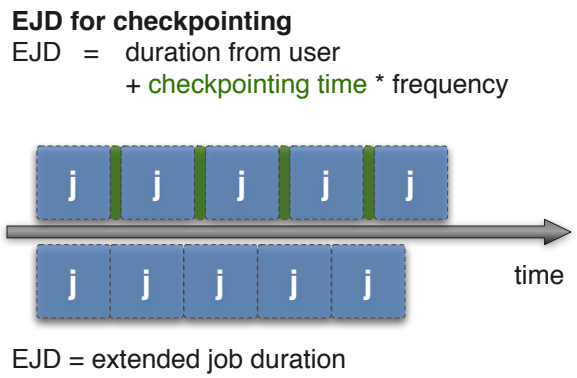


Figure 6.4: EJD Resulting from Checkpointing Initialisation

If the provider is only able to generate checkpoints, but not to perform a migration, the snapshot data can be forwarded to another provider supporting migration. The external job completion counts among outsourcing and, in the context of a commercial Grid environment, the original provider will negotiate an SLA for the outsourcing. The SLA negotiation enables the provider to define resource and time constraints based on the initial SLA for the job which has been accepted from the service consumer. Note that the provider could initiate outsourcing even if the job has not been started yet. Outsourcing the job before its start is beneficial, if it is likely that not enough resources will be available to perform the job according to the time

and resource constraints. In a commercial Grid environment restrictions can be defined from service consumer regarding providers which may be used for outsourcing. Such limitations are not considered in this work since these can be handled by appropriate policies.

If the underlying RMS is not able to perform checkpointing, a job restart can be prevented if the job is executed redundantly. A second instance of the execution implies that a job requesting for n nodes needs $2n$ nodes. Obviously, the number of parallel initiated instances of the job execution is only limited by the resources of the provider. However, each instance results in a multitude of internal costs for the resource utilisation and thereby reduces the provider's profit.

Resource outages often result in an SLA violation if no FT-mechanism is initiated from the RMS. The most profitable and promising approach to prevent an SLA violation is to use checkpointing and migration. In a planning based system the initialisation of checkpointing should be planned during the SLA negotiation since the job duration has to be extended. In addition to these capabilities, outsourcing the job to another provider is possible for jobs independent from their job state (not started/no checkpoint, with checkpoint). If the underlying RMS does not support checkpointing, the RMS can plan several instances of the job execution. The instances can be started to the same or different times and are performed in scope of a redundant job execution.

Each initiated FT-mechanism results in cost since either resources are used longer, additional storage is needed, revenues have to be paid to other providers or more compute nodes are used. Whereas the extended job duration and the additional storage is often negligible, a redundant job execution results in significant higher costs for the provider since a multitude of the resources requested are used.

6.2 Risk Assessment

The main issue in risk assessment is the correct determination of the rate of occurrence of an event because statistical information may not be available on all kinds of past incidents. This has turned out to be a significant problem for Grid environment as statistics on (for instance) node failures or the failure of a node's components is not collected routinely and the most important basis for assessing *systematic* risk is not available. In contrast to static risk assessment which is allowed to rely on statistics, dynamic risk assessment is to be done on-line and in real time. Thus the amount of empirical data available is significantly smaller in order to be able to react to recent changes of resource behaviour. Risk assessment can be performed for a short planning period. If working with smaller data sets, it is necessary to cope with the real variations in node failures, task arrivals, and random maintenance events; in larger (or very large) data sets these variations can be expected to be smoothed out. As a data basis for developing the dynamic risk assessment model, monitoring data of the Grid'5000 has been used which was published in scope of [Iosu 07].

This section presents the risk assessment model developed which can be applied to estimate PoFs of SLAs in planning as well as queueing based systems. The generality enables to re-use the model also in RMS following different strategies and policies. Before detailing the model,

Section 6.2.1 introduces the framework and considerations which have lead to the dynamic model as presented in Section 6.2.2. Details to the dynamic risk assessment can be found in [Voss 08b].

6.2.1 Introduction

The models for the dynamic risk assessment is based on the theory of stochastic processes. The main reason for this is the lack of adequate data on node performance, node failures, and resource availability for clusters and Grids. If the data would be sufficient and reliable, it would be possible to build a family of causal models for different types of nodes, clusters, and Grids in order to *explain* and *predict* the future behaviour of the nodes, clusters, and Grid. If the mechanisms driving the failures of nodes are not sufficiently well known to build causal models and/or it is not possible to combine explanations of node failures to explain the performance of clusters or the Grid in general, the future performance of nodes (and clusters and Grids) can be *described* and *predicted* with the use of the time series models [Rama 95, Box 90, Alex 01, Elli 06, Tsay 05].

As an overall approach to dynamic risk assessment the models used in Baysian statistics are pretty useful. In Baysian statistics it is started from some prior distribution of failure rates, based on the offered PoFs. Consider the simple case, where the offered PoF and the true PoF are the same for all SLAs. Now suppose that when the offered PoF is $1 - P_{\text{offered}}$, then the true PoF is developed from the distribution:

$$P\left(P_{\text{true}} = P_{\text{offered}}\right) = \frac{1}{3} \tag{6.1}$$

$$P\left(P_{\text{true}} = P_{\text{offered}} + \delta P_{\text{offered}}\right) = \frac{1}{3} \tag{6.2}$$

$$P\left(P_{\text{true}} = P_{\text{offered}} - \delta P_{\text{offered}}\right) = \frac{1}{3} \tag{6.3}$$

Here δ is a small constant and a discrete distribution is chosen only for simplicity. This could be replaced by a continuous distribution, which is the approach used for the dynamic risk assessment. Now assume that, from N SLAs negotiated F failed. Recall that Bayes' theorem can be stated as:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|\overline{B})P(\overline{B})} \tag{6.4}$$

Let event A correspond to F SLA violations/failures being observed and event B be that the particular PoF under consideration is the correct one. Thus, $P(B) = \frac{1}{3}$ for each possible true PoF according to the distribution in equations (6.1)-(6.3), where $P_{\text{t}} := P_{\text{true}}$ and $P_{\text{offered}} :=$

P_{O}:

$$P\left(F \text{ fails} | P_{\mathrm{t}} = P_{\mathrm{O}}\right) = \binom{N}{F} \left(P_{\mathrm{O}}\right)^{F} \left(1 - P_{\mathrm{O}}\right)^{N-F} \tag{6.5}$$

$$P\left(F \text{ fails} | P_{\mathrm{true}} = P_{\mathrm{O}} + \delta P_{\mathrm{O}}\right) = \binom{N}{F} \left(P_{\mathrm{O}} + \delta P_{\mathrm{O}}\right)^{F} \cdot \left(1 - P_{\mathrm{O}} - \delta P_{\mathrm{O}}\right)^{N-F} \tag{6.6}$$

$$P\left(F \text{ fails} | P_{\mathrm{t}} = P_{\mathrm{O}} - \delta P_{\mathrm{O}}\right) = \binom{N}{F} \left(P_{\mathrm{O}} - \delta P_{\mathrm{O}}\right)^{F} \cdot \left(1 - P_{\mathrm{O}} + \delta P_{\mathrm{O}}\right)^{N-F} \tag{6.7}$$

$$\tag{6.8}$$

$$\begin{aligned} \Rightarrow P\left(F \text{ fails} | P_{\mathrm{t}} \neq P_{\mathrm{O}}\right) &= \frac{1}{2}\left[\binom{N}{F} \left(P_{\mathrm{O}} + \delta P_{\mathrm{O}}\right)^{F} \cdot \left(1 - P_{\mathrm{O}} - \delta P_{\mathrm{O}}\right)^{N-F}\right] \\ &+ \frac{1}{2}\left[\binom{N}{F} \left(P_{\mathrm{O}} - \delta P_{\mathrm{O}}\right)^{F} \cdot \left(1 - P_{\mathrm{O}} + \delta P_{\mathrm{O}}\right)^{N-F}\right] \end{aligned} \tag{6.9}$$

These expressions allow to derive $P(P_{\mathrm{true}} = \text{whatever} | F \text{ fails})$ using Bayes' theorem. The calculated PoF can be described as the mean value of the PoF according to the new distribution. If the end-user requests, the new distribution itself can be provided as additional information.

In general the Baysian Data Analysis [Gelm 03] is a statistical process which is used to estimate parameters of an underlying distribution based on the observed distribution. The observed distribution is denoted prior distribution and defined as follows:

Definition 6.2.1 (Prior Distribution [Gelm 02])
The *prior distribution* is a key part of Baysian inference (see Baysian methods and modelling) and represents the information about an uncertain parameter θ that is combined with the probability distribution of new data to yield the posterior distribution, which in turn is used for future inferences and decisions involving θ.

After a prior distribution is defined, which is completely arbitrary and can also consists of assessed likelihoods of parameters, data is collected to obtain the observed distribution, i. e. here the PoF distribution. In the next step the likelihood of the observed distribution is calculated as a function of parameter values. Afterwards this likelihood function is multiplied with the prior distribution and normalised in order to ensure that probabilities are in the interval [0,1]. The result is called *posterior distribution*. For more information see [Sivi 96, Hoel 71, Iver 84].

In order to estimate the PoF of SLAs, it is necessary to consider resource stabilities and availabilities for which the Baysian Data Analysis is useful. Based on the probability of a resource failure in a given time interval, i. e. the execution time of a job, the PoF for an SLA can be determined. The estimation of the probability of an SLA violation has to also take into account the availability of alternative resources which may be used to compensate for resource failures. As a consequence, FT-mechanisms or strategies of the resource management influence the PoF of SLAs which differs those from PoF of resources. Several factors are of importance when considering the uncertainty about the expected amount of resources available at a given time point. These are listed as follows:

1. Computing tasks arrive at a heterogeneous rate to the Grid. Thus, when assessing the uncertainty related to the Grid workload within a specific time interval, it is necessary to take into account differences in day/night and weekdays/weekend patterns. Depending on the administration characteristics of the actual system described, it may be relevant to include the queuing system explicitly in the statistical model if not a planning-based scheduler is used. However, in the following the assumption is made that the number of computing tasks waiting for their execution at a particular point time, say t, is *not* explicitly conditioned on when calculating the relevant system probabilities at time $t + s, s \geq 0$.
2. Computing tasks have varying characteristics with respect to the number of nodes utilised and the length of the execution time. Initially, these two random quantities can be considered as conditionally independent, which leads to a simpler statistical model. Nevertheless, it would be important to attempt to verify such an assumption empirically, as a moderate to strong dependence between the number of nodes utilised and the length of the execution time is expected to induce a considerable bias in the uncertainty assessment.
3. The availability of Grid resources (and the success of a computing task) is limited by two types of events. Firstly, technical failures of nodes, either individual or groups of nodes, can lead to a premature termination of a task, unless some reserve resources are available which may be used at the time of the failure. Secondly, maintenance work on a cluster affects the resource availability. When considering the risks associated with a premature termination or a failure of a computing task, the characteristics of the maintenance works have to be appropriately formulated. One crucial aspect is whether the system managers have the policy to shut down parts of clusters (or whole clusters), even in the presence of ongoing computation tasks, or whether it is the policy to await until the completion of such tasks before initiating maintenance. To be as general as possible, node failures and maintenance events are separately modelled.

6.2.2 Dynamic Risk Assessment Model

Since a RMS is responsible for only one cluster, in the following only the resource outages of nodes on one cluster are considered. However, it is easy to generalise this in order to estimate resource availabilities on a Grid site or generally in the Grid. Furthermore the model is able to be used in queuing and planning based systems since arrival rates of jobs are taken into account. In planning based systems the arrival rates are however only interesting for predicting the workload for the execution time since it plans directly after it has received the job request. Detailed information about distributions referred or used in this section can be found in [Evan 00].

Let the cluster consists of n nodes and let $\lambda(t) > 0, t \geq 0$ denote generally a time-nonhomogeneous rate function for a Poisson process $N(t)$, several types of which will be used to characterise the resource availability. Initially, the time unit defining a utilisation is 1 minute and the time window equals one week, from Monday 00 am to Sunday 12 pm. This enables the rate function to be defined such that $\lambda(t)$ depends on both the hour of a day and the day of the week, however, such that anticipated exchangeabilities during these periods can be taken into account in the estimation of $\lambda(t)$. Should the initially chosen time unit turn

out to be suboptimal, the models developed below can still be applied as such, by a simple modification of the rate function.

The rate function specifies the expected number of events in any given time interval $(t_1, t_2]$ according to $\lambda_{t_1,t_2} = \int_{t_1}^{t_2} \lambda(t)dt$, and the probability distribution for the number of events $X = N(t_2) - N(t_1)$ equals

$$p(X = x) = \frac{e^{-\lambda_{t_1,t_2}}(\lambda_{t_1,t_2})^x}{x!}, x = 0, 1, \ldots \tag{6.10}$$

Thus far three distinct types of events have been defined (computing task arrival, maintenance arrival, and node failures) that occur in the system. A separate rate function can be defined for each of these event types, denoted by $\lambda^1(t), \lambda^2(t), \lambda^3(t)$, respectively. Note that, if the arrivals of the different types of events are assumed independent, then the sum of such events will be governed by a Poisson process with the rate function $\lambda^1(t) + \lambda^2(t) + \lambda^3(t)$.

Depending on the characteristics of the Grid, it may be necessary to consider the workload, failure and maintenance events separately for each cluster, if the system consists of at least moderately heterogeneous components in this respect. It is quite likely, that at least failure rates vary considerably over the clusters, which would motivate a model with separate failure rate parameters in order to allow a customisation of the model to various clusters/providers. However, to simplify the model structure, the failure rate can be treated as a constant over the time window considered here, i. e. $\lambda^3(t) = \lambda^3$, for all $t \geq 0$.

For the estimation of the Poisson rate parameters *a priori* information may be utilised to specify homogeneous segments of time where the rate parameters can be assumed constant. Using the information from the Grid'5000 study [Iosu 07], it is anticipated that the arrival intensity of tasks during the daytime on weekdays is relatively homogeneous, and a similar condition holds for the night hours. Moreover, hours during weekends can be treated in an analogous day/night fashion. If the time interval $(t_1, t_2]$ corresponds to such a homogeneous segment with x observed events, the likelihood function for the rate parameter is given by

$$p(x|\lambda_{t_1,t_2}) = \frac{e^{-\lambda_{t_1,t_2}}(\lambda_{t_1,t_2})^x}{x!}. \tag{6.11}$$

By collecting data (counting of events) $x_1, \ldots, x_r$ from r comparable time segments, the joint likelihood function is obtained

$$p(x_1, \ldots, x_r|\lambda_{t_1,t_2}) \propto e^{-r\lambda_{t_1,t_2}}(\lambda_{t_1,t_2})^{\sum_{i=1}^r x_i}. \tag{6.12}$$

By using a Gamma(α, β) prior distribution for the rate parameter λ_{t_1,t_2}, the posterior distribution will be a Gamma$(\alpha + \sum_{i=1}^r x_i, \beta + r)$ distribution and it has the density

$$p(\lambda_{t_1,t_2}|x_1, \ldots, x_r) \propto e^{-(r+\frac{1}{\beta})\lambda_{t_1,t_2}}(\lambda_{t_1,t_2})^{\alpha-1+\sum_{i=1}^r x_i}. \tag{6.13}$$

Also, under a limiting reference prior ($p(\lambda_{t_1,t_2}) \propto \lambda_{t_1,t_2}^{-1}$), the Gamma form still holds for the posterior. The predictive distribution of the number of events is a Poisson-Gamma-distribution, obtained by integrating the likelihood with respect to the posterior. Under the

reference prior the predictive probability of having x events in future on a comparable time interval equals

$$p(x|r, \frac{1}{2} + \sum_{i=1}^{r} x_i, 1) = \frac{\Gamma(\sum_{i=1}^{r} x_i + x + 1/2) r^{\left(\frac{1}{2}+\sum_{i=1}^{r} x_i\right)}}{\Gamma(1/2 + \sum_{i=1}^{r} x_i) x! (r+1)^{\left(\frac{1}{2}+\sum_{i=1}^{r} x_i\right)+x}}. \tag{6.14}$$

This distribution has the mean $(1/2 + \sum_{i=1}^{r} x_i)/r$ and the variance $\sum_{i=1}^{r} x_i(r+1)/r^2$. Details about the computation process are presented in Section 6.2.2.1.

When a computing task begins its execution, its successful completion requires a certain number of nodes to be available over a given period of time. To assess the uncertainty about the resource availability, it is necessary to model both the distribution of the number of nodes and the execution time required from the task. The most adaptable choice of a distribution for the number of nodes required, say M, is the multinomial distribution

$$p(M = m) = p_m, m = 1, \ldots, u, \tag{6.15}$$

where u is an *a priori* specified upper limit for the number of nodes. To model resource outages and unavailability caused by maintenance a similar distribution can be defined. Such a vector of probabilities will subsequently be denoted by $\mathbf{p}$. For example u could equal the total number of nodes of the cluster, however, such a choice would lead to an inefficient estimation of the probabilities, and therefore, the upper bound value should be carefully assessed using empirical evidence.

An advantage of the use of the multinomial distribution in this context is its ability to represent any type of multimodal distributions for M, in contrast to the standard parametric families, such as the Geometric, Negative Binomial and Poisson distributions. For instance, if there are two major classes of computing tasks or maintenance events, such that one class is associated with relatively small numbers of required nodes, and the other with relatively large numbers, the system behaviour in this respect can well representable by a multinomial distribution. On the other hand, standard parametric families of distributions would not enable an appropriate representation, unless some form of mixture distribution were utilised. Such a choice would complicate the inference about the underlying parameters due to the fact that the number of mixture components would be unknown *a priori*.

A disadvantage of the multinomial distribution is that it contains a large number of parameters when u is large. However, this difficulty is less severe when the Baysian approach to parameter estimation is adopted. Given observed data on the number of nodes required by computing tasks, the posterior distribution of the probabilities $\mathbf{p}$ is available in an analytical form under a Dirichlet prior, and its density function can be written as

$$p(\mathbf{p}|\mathbf{w}) = \frac{\Gamma(\sum_{m=1}^{u} \alpha_m + w_m)}{\prod_{m=1}^{u} \Gamma(\alpha_m + w_m)} \prod_{m=1}^{u} p_m^{\alpha_m + w_m - 1}, \tag{6.16}$$

where w_m corresponds to the number of observed tasks utilising m nodes, α_m is the *a priori* relative weight of the mth component in the vector $\mathbf{p}$, and $\mathbf{w}$ is the vector $(w_m)_{m=1}^{u}$. The corresponding predictive distribution of the number of nodes required by a generic computing

task in the future equals the Multinomial-Dirichlet distribution, which is obtained by integrating out the uncertainty about the multinomial parameters with respect to the posterior distribution. The Multinomial-Dirichlet distribution is in our notation defined as

$$\begin{aligned} p(M = m^*|\mathbf{w}) = & \frac{\Gamma(\sum_{m=1}^{u} \alpha_m + w_m)}{\prod_{m=1}^{u} \Gamma(\alpha_m + w_m)} \frac{\prod_{m=1}^{u} \Gamma(\alpha_m + w_m + I(m = m^*))}{\Gamma(1 + \sum_{m=1}^{u} \alpha_m + w_m)} \\ = & \frac{\Gamma(\sum_{m=1}^{u} \alpha_m + w_m)}{\Gamma(\alpha_{m^*} + w_{m^*})} \frac{\Gamma(\alpha_{m^*} + w_{m^*} + 1)}{\Gamma(1 + \sum_{m=1}^{u} \alpha_m + w_m)}. \end{aligned} \tag{6.17}$$

If a more elaborate model is needed, it would be possible to induce a neighbourhood structure among the elements of $\mathbf{p}$, by a smoothing hyperprior, such that a relatively high value for the probability of observing any particular value m would imply higher values also to the immediately adjacent values $m - 1$ and $m + 1$. However, when large quantities of data are available for the estimation of $\mathbf{p}$, the need for any advanced smoothing techniques is expected to be negligible.

By combining the above distributions, the probability distribution may be derived for the number of nodes in use for computing tasks in a future time interval $(t_1, t_2]$, as the corresponding random variable equals the product XM. As noted earlier, the analogous models for the maintenance effects and the node failures are similarly derived. Such a product random variable can be combined with a model for the anticipated length of the use of the resources.

To simplify the inference about the execution time of a task affecting a number of nodes, initially the length follows a Gaussian distribution with expected value μ and variance σ^2. Obviously, it is motivated to have separate parameter sets for different types of tasks. Assume now that data $t_1, \ldots, t_n$ representing the lengths (in minutes) of n tasks is available. This leads to the sample mean $\bar{t} = \sum_{i=1}^{n} t_i$ and variance $s^2 = n^{-1} \sum_{i=1}^{n} (t_i - \bar{t})^2$. Assuming the standard reference prior for the parameters, the predictive distribution for the length of a future task, say T, is obtained which has the T-distribution with parameters $\bar{t}, ((n-1)/(n+1))s^2, n-1$, i. e. the probability density of the distribution equals

$$p\left(t|\bar{t}, \left(\frac{n-1}{n+1}\right) s^2, n-1\right) = \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\Gamma(1/2)} \left(\frac{1}{(n+1)s^2}\right)^{\frac{1}{2}} \left[1 + \frac{1}{n+1)s^2} (t - \bar{t})^2\right]^{-\frac{n}{2}}. \tag{6.18}$$

The probability that a task lasts longer than any given time t equals $P(T > t) = 1 - P(T \leq t)$, where $P(T \leq t)$ is the *Cumulative Density Function* (CDF), i. e. probability distribution function, of the T-distribution. The value of the CDF can be calculated numerically using existing functions. However, it should also be noted that for a moderate to large n, the predictive distribution is well approximated by the Gaussian distribution with the mean $\bar{t}$ and the variance $s^2 \frac{(n+1)}{(n-3)}$. Consequently, if the Gaussian approximation is used, the probability $P(T \leq t)$ can be calculated using the CDF of the Gaussian distribution.

In the next step the probability that a computing task is successful has to be estimated. This happens if there will always be at least a single idle node available in the system in the case of a node failure. Let $S = 1$ denote the event that the task is successful and $S = 0$ the opposite event. The probability of the success is formulated as the sum $P(\text{"none of the nodes allocated to the task fail"}) + \sum_{m=1}^{m_{\max}} P(\text{"}m \text{ of the nodes allocated to the task fail \& at least } m \text{ idle}$

nodes are available as alternatives"). Here $m_{\max}$ is an upper limit for the number of failures considered. The value can be chosen by judging the size of the contribution of each event, determined by the corresponding probability. Thus, the sum may be simplified by considering only those events that do not have vanishingly small probabilities. Notice that a simplification is performed for the events below by considering the m failures to take place simultaneously. This results in:

$$\begin{aligned} P(S=1) &= 1 - P(S=0) \\ &= 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur \& less than } m \text{ free nodes available}) \\ &= 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) P(\text{less than } m \text{ free nodes available}) \end{aligned}$$

$$\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) P(\text{less than } m \text{ free nodes at any time point}) \tag{6.19}$$

The probability $P(m$ failures occur) is directly determined by the failure rate model discussed above. The other term, the probability P(less than m free nodes at any time point), on the other hand, is dependent both on the scheduler features for job allocation and the need of reserve nodes by the other tasks running simultaneously. Thus, the failure rate model will be used to calculate the probability distribution of the number of reserve nodes that will be jointly needed by the other tasks (that are using a certain total number of nodes) during the computation time that has the distribution specified above for a single node.

When appropriate empirical data becomes available, it is possible to fit the proposed model for resource availability and assess its validity. Given the time-ordered characteristic of the data, an appropriate strategy to validation is to exclude a certain sequence of time points from the data for validating purposes. The validation can be numerically performed by computing the predictive distribution function for the quantities of interest and comparing this directly to the empirical distribution function calculable from the excluded data.

This model is used in order to determine the initial PoF for an SLA. If specific Risk Management activities are planned during the SLA negotiation, the PoF is reduced. Chapter 7 presents enhancements of this initial PoF which take into account such Risk Management strategies. For easier implementation we have worked out the logarithmic version of equation (6.14) with the help of a recurrence formula. The following section present details.

6.2.2.1 Computation of Equation 6.14

Equation 6.14 can be computed as described in the following. It conforms to

$$\Gamma(n+1/2) = \frac{1 \times 3 \times \cdots \times (2n-1)}{2^n} \times \sqrt{\pi} \tag{6.20}$$

where $y = x_1 + \ldots + x_r$. By using the equality

$$\Gamma(n + 1/2) = \frac{1 \times 3 \times \cdots \times (2n-1)}{2^n} \times \sqrt{\pi} \tag{6.21}$$

is possible to determine

$$\frac{\frac{1 \times 3 \times \cdots \times (2(x+y)-1)}{2^{x+y}} \times \sqrt{\pi} \times r^{y+1/2}}{\frac{1 \times 3 \times \cdots \times (2y-1)}{2^y} \times \sqrt{\pi} \times x! \times (r+1)^{x+y+1/2}} \tag{6.22}$$

which turns into

$$\frac{(2y+1) \times (2y+3) \times \cdots \times (2(x+y)-1) \times r^{y+1/2}}{2^x \times x! \times (r+1)^{x+y+1/2}} \tag{6.23}$$

Taking the natural logarithm of both sides, results in

$$\begin{aligned} \ln p(x|r, x+1/2, 1) = \quad & \ln(2y+1) + \ln(2y+3) + \cdots + \ln(2y+2x-1) \\ & + (y+1/2)\ln r - x\ln 2 - \ln x! - (x+y+1/2)\ln(r+1) \end{aligned} \tag{6.24}$$

which conforms to:

$$\begin{aligned} \ln p(x|r, x+1/2, 1) &= \ln(2y+1) + \ln(2y+3) + \cdots + \ln(2y+2x-1) \\ &+ (y+1/2)\ln r - x\ln 2 - \sum_{i=1}^{x} \ln i - (x+y+1/2)\ln(r+1) \end{aligned} \tag{6.25}$$

Thus the computation of the conditional probability can be summarised as follows:

- r is the number of the time intervals from which we take the data (comparable as length)
- $x_1, \ldots, x_r$ are the data (where x_i represents the number of tasks arrived in the i-th interval)
- x is the predictable number of tasks which will arrive in a time interval in the future having the length comparable with the other r intervals.

First the natural logarithm of conditional probability $\ln p(x|r, x+1/2, 1)$ is computed and then, the conditional probability from the equation $p(x|r, x+1/2, 1) = e^{\ln p(x|r, x+1/2, 1)}$.

6.3 Procedure of Defining Risk Management Processes

Integrating Risk Management processes in SLA provisioning can be divided into two different phases: during and after SLA negotiation (see Figure 6.5). A differentiation is meaningful since before accepting an SLA, providers only evaluate whether they should commit or reject. Hence, in the first phase the provider's main task is to evaluate the risk of SLA acceptance and decide to either accept or avoid this risk. Risk Management activities primarily support the decision making and if necessary risk reduction can be performed if the risk is too high for

the provider. After the SLA has been committed from both parties – service consumer and provider – the situation changes significantly since the provider will definitely lose profit if it does not provide the service as negotiated. Hence, the objective is to not violate any SLA. If however a critical number of resources have failed, not all SLA fulfillments can be achieved. In this case providers have to act carefully and make decisions under the consideration of PoFs since at this stage a job cannot be considered without its impact for other jobs. Risk Management can support to find the in expectation most profitable solution.

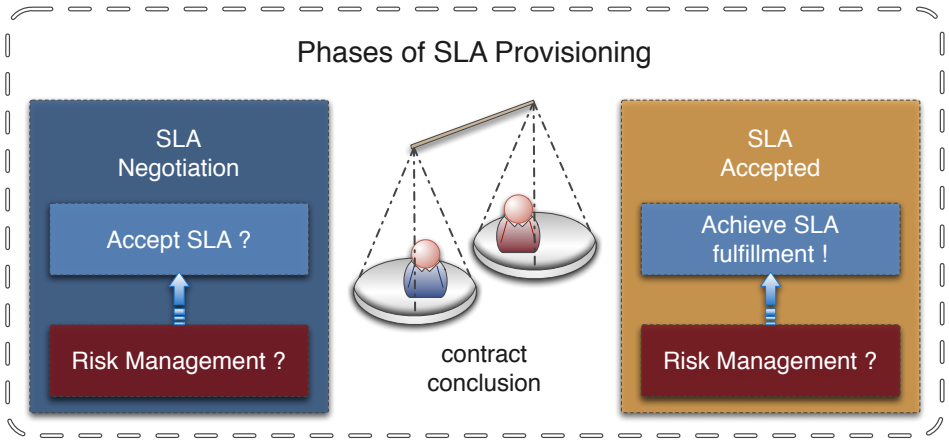


Figure 6.5: Phase Differentiation in SLA Provisioning

The questions to be answered in the following chapters concern the opportunities to support these main tasks by Risk Management processes. *Which activities can be initiated in scope of risk treatment? How can these be classified as risk acceptance, risk avoidance, risk transference, and risk mitigation/reduction? Which strategies can be followed in the decision process of the Risk Management ? When is the general Risk Management process and when the targeted Risk Management process applicable?*

Following in Chapter 7 the SLA negotiation is examined and the Risk Management integrable in this phase is presented. Risk Management activities in the post-negotiation phase are described in Chapter 8.

Chapter 7

Risk Management During SLA Negotiation

The first starting point to integrate risk awareness from the perspective of SLA provisioning is the SLA negotiation. It is not only the previous step before the job execution, also does it form the basis for considering failure probabilities in SLAs (cf. Figure 7.1). This chapter replies how to integrate Risk Management in this first phase to support the provider's decision making and evaluation of accepting a received SLA request.

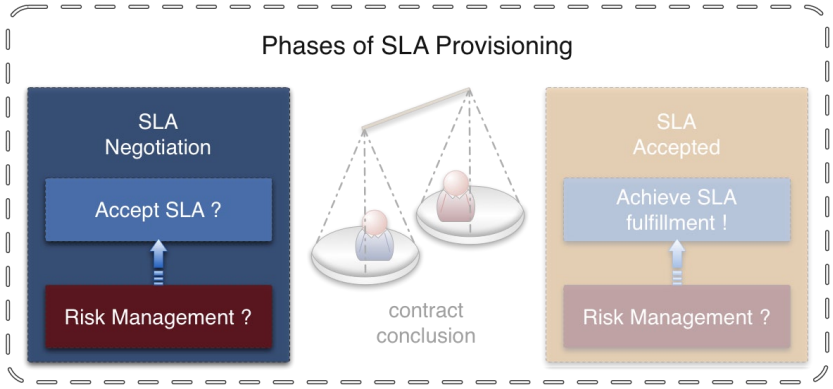


Figure 7.1: How to Benefit from Risk Management During SLA Negotiation ?

Since the underlying model defines the usage of a planning based system, the scheduler makes an advance reservation during the SLA negotiation which will be used for the job execution after the contract conclusion. Section 7.1 points out that considering PoFs during this resource reserving is the first step of introducing risk awareness in the Grid fabric. Section 7.2 presents a scheduling mechanism for advance reservations by focusing on PoFs. Since often providers want to benefit from risk awareness as an additional value and not to replace their strategies and policies completely, in Section 7.3 coupling arbitrary scheduling mechanisms with risk awareness is described.

If the PoF feasible by using adequate available resources is too high, the Risk Management strategy of risk reduction can be followed. During SLA negotiation there exist various mechanisms to reduce the risk for a provider. Section 7.4 presents how to mitigate risks in order to accept an SLA. The applicability and meaning of other Risk Management strategies – risk acceptance, risk avoidance, and risk transferring – in the negotiation phase are examined in Section 7.5.

According to the WS-Agreement Negotiation (WSAN) different SLA requests and offers can be exchanged between providers and consumers in scope of renegotiations. The advance reservation made after receiving the first SLA request can be reused, has to be slightly adjusted, or has to be discarded (and a new reservation has to be made). A brief overview of the reutilisation of the reservations of the first negotiation phase for following negotiation steps is shown in Section 7.6. Section 7.7 completes this chapter by recapitulating the Risk Management activities and ideas presented in Sections 7.2 and 7.3 in order to answer the integration of Risk Management during the SLA negotiation as stated out in the question in Figure 7.1.

7.1 Purpose of Integrating Risk Management During Negotiation

Considering failure probabilities during the SLA negotiation is beneficial for providers as well as service consumers. As pointed out in [Voss 07c], providers can gain the users' trust by publishing the PoF of SLAs, leading to an increase of users' confidence once these probabilities are clearly stated.

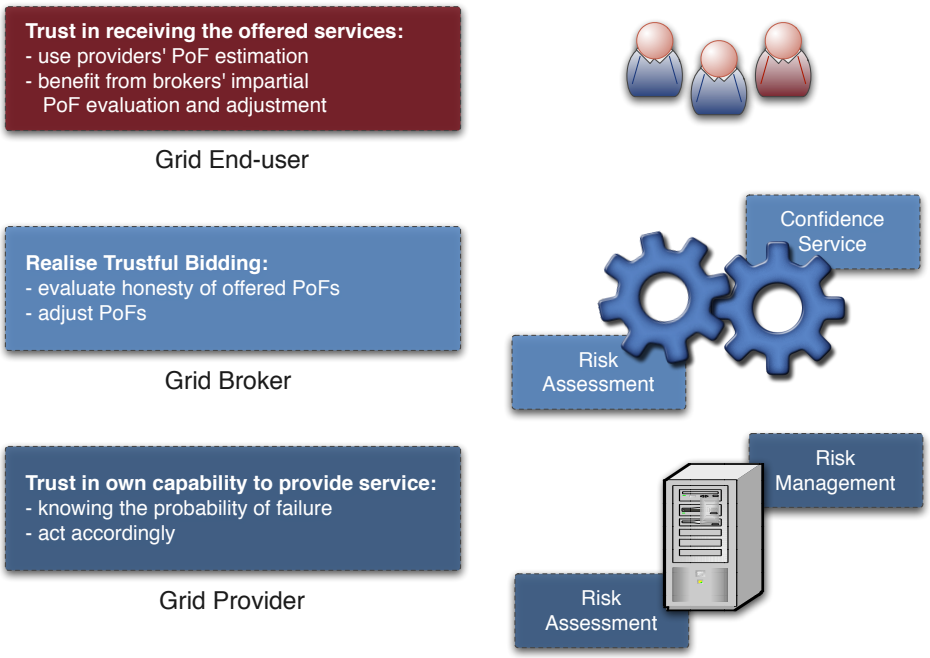


Figure 7.2: Increasing Trust By PoFs [Voss 07c]

Service consumers can use this information as additional information in the SLA negotiation process. A Grid broker can ensure provider's honesty by generating statistics about the reliability of its PoF estimations. Based on these statistics, the broker can adjust the PoF of the provider if this is marked as unreliable. [Gour 08] presents details of this concept realised through a module denoted *confidence service*. Consequently, the idea of integrating PoF estimations in SLAs seems to be feasible in the Grid and beneficial for Grid commercialisation.

Figure 7.2 summarises the coherent concept of using and benefiting from the PoF as an additional SLA parameter from the perspective of increasing trust for all Grid actors.

The resources used for providing the requested service have an high impact for the successful SLA provisioning since a resource outage is the main threat of an SLA violation. In planning based systems the resource mapping is already generated during the SLA negotiation and thereby before the contract conclusion. This forms the basis for negotiating about failure probabilities on a per job basis instead of general PoFs offered according to quality standards.

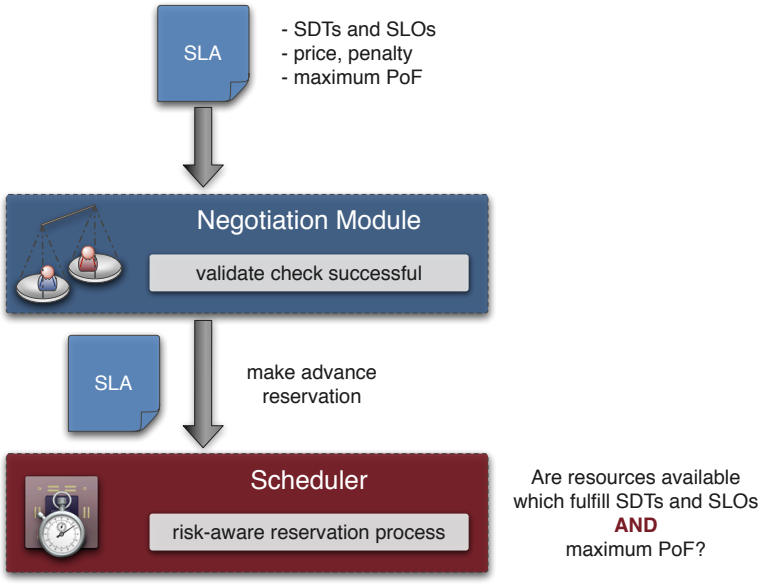


Figure 7.3: Risk Management Addresses Resource Reserving during Negotiation

In a risk aware Grid, service consumers define an upper bound for the PoF they are willing to accept. The provider may modify this PoF by means of a counteroffer, however, in most cases this is only meaningful if the provider offers for the same price a better service, i. e. a lower PoF. In the following, the assumption is made that the consumer has defined a maximum PoF in their SLA request. This is no limitation since consumers could define an upper bound of 100% if they are willing to accept any probability. In order to offer accurate PoFs or compare a feasible PoF with the maximum value, the provider has to consider the PoF during the initial reservation process. Figure 7.3 depicts the addressed step to integrate risk awareness. According to the RMS's workflow of handling job requests during the negotiation (see Section 6.1.2), the negotiation module has received an SLA request and has checked its validity. If the check was passed successfully, the negotiation module commissions the scheduler to make an advance reservation according to the SLA request. The challenge in a risk aware context is not only to find resources conforming to the *Service Description Terms* (SDTs) and *Service Level Objectives* (SLOs), rather to estimate the feasible PoF and set it in relation to the maximum accepted value. Consequently, this risk consideration has to be integrated in the resource reserving process.

The maximum accepted PoF is initially determined by the service consumer and part of

the SLA request. The negotiation module can adjust this upper bound during the check of validity. A modification might be necessary because of consumer or internal policies. For example a provider follows the strategy to make offers with a PoF $x\%$ lower than the maximum accepted. The lower PoF may be published or only considered internally in the resource management. If the feasible PoF is published, this either is realised by a counteroffer in WS-Agreement Negotiation or available by request of the consumer. If the modified PoF is only internally considered, the scheduler does not know the actual upper bound and works with the PoF defined by the negotiation module instead.

Another reason for a modification might be that a provider's policy defines a range of values which are acceptable for itself. For example, a provider always want to offer SLAs having a PoF between 5% and 15%. Such a range can be determined if the service provisioning should follow specific quality constraints or limitations. Since the PoF offered is usually reflected in the penalty fee, policies might determine which penalty fees are allowed to accept.

Note that not only considering the PoF during the reservation process is one aspect to be solved. To coupling Risk Management with the processes during the SLA negotiation, a consideration is necessary of the applicability of the risk treatment strategies:

- risk acceptance,
- risk avoidance,
- risk transference, and
- risk mitigation/reduction.

7.2 Focusing on PoF in Resource Reserving

The resource reservation process during the SLA negotiation is performed by the scheduler. It plans the job execution according to the SLA request defining SDTs, SLOs, and a maximum accepted PoF. Consequently, the scheduling should focus on reserving resources in that way that the maximum value is fulfilled. If a lower feasible PoF is not published, the provider should target to make a reservation resulting in a PoF lower than the maximum but close to this value. If the feasible PoF is published, it might be advantageous to make a reservation with the lowest PoF which is possible. However, this has two disadvantages: first, the lower the PoF, the higher the revenue. Hence, if offering the lowest feasible PoF, the increased revenue might not being worth for the consumer under consideration of the PoF and she will not accept the bid. Secondly, offering the lowest feasible PoF would lead to a prioritisation of earlier arriving jobs and later arriving job requests cannot be accepted because of the previous reservations. I.e. if the reservation for job j achieves the lowest PoF, this can be provided for job j, but if after the agreement of j a job request j' with similar requirements is received asking for a lower PoF than j did, this PoF might not be feasible to fulfill.

A reservation, which results in the most adequate PoF, might result in a bad schedule with more or probably unusable gaps in which the resources will be idle and no other job can be executed. Consequently, even if the focus of the scheduling process is on achieving the maximum PoF, the resultant schedule has to be also considered. This section addresses how to balance both aspects. In [Hove 06c] these ideas have been published for single jobs.

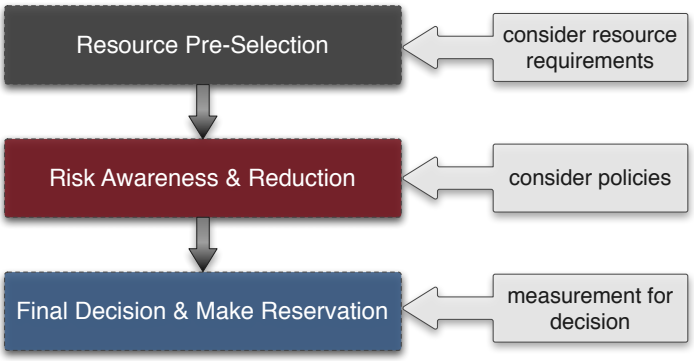


Figure 7.4: Reservation Process Focusing on PoF – Overview

The reservation process can be divided into three main phases as depicted in Figure 7.4: first of all a resource pre-selection is made as described in Section 7.2.1. Before determining the final decision whether and which reservation to use for the job execution (see Section 7.2.3), Section 7.2.2 presents how the Risk Management purpose is addressed. The risk awareness is influenced from policies. The resource reservation process focusing on PoFs utilises measurements in the decision making to compare different possibilities. These can be defined according to the provider's policies, examples are presented in Section 7.2.4.

7.2.1 Resource Pre-selection

To make an advance reservations for an SLA bound job, not every time slot is suitable in the schedule and not every resource can be used. All SLOs of the SLA have to be fulfilled by a reservation since otherwise the provider knows a priori that it would violate the SLA. Consequently, it would not accept the SLA in this case.

The scheduler's reservation process has to find resources which comply with the PoF restrictions, policies determined by the negotiation module, and conform to all SLOs of the SLA. In order to not only consider the PoF and also keep possibilities open for later arriving job requests, the modification of the schedule quality has to be taken into account. In particular, the reservation process is the initial step of the job execution and at this stage the scheduler has the highest flexibility if several execution slots conform to the SLA's time constraints. If available, this flexibility should be utilised to generate a schedule with a minimum of interspaces between the job execution. Accordingly, the quality of the schedule is in the prime focus if several reservations can be made which fulfill the PoF requested. As a matter of course, the resources selected for the job execution have to possess a free time slot and also to dispose of the resource requirements (number of CPUs, performance, software, connections, etc.).

To find resources conforming to all SLA requirements (including the PoF) and also resulting in a good schedule, a pre-selection of resources, which may be reserved for the job execution, is the initial step. The assumption is made that the SLA bound job j asks for n compute nodes for a time period d. The scheduler identifies possible execution slots in which n resources are free. Those resources are grouped in a set $E_{i|\{s,e\}}$ where i is a running number and the possible execution window is defined by s as the starting point and e as the endpoint. These

candidates $E_{i|\{s,e\}}$ are hold in a set R. The time points s and e have to conform to the earliest start time, duration, and deadline defined in the SLA request by the contractor. Note that using the running index i is caused by the fact that several resource sets may have the same execution window s, e. A differentiation of such resource sets might exist if the resources are reserved explicitly, in the context of reserving mapping this means that jobs are assigned to be executed on specific resources in place of to a number of resources. If resource stabilities are not similar for all resources on the same cluster, a differentiation is also reasonable.

If job j only needs *one* compute node, the definition of the planned execution time has often many possibilities. However, this phenomena is also possible for parallel jobs using multiple nodes. From the resource-driven perspective a resource or a group of resources can have several free time slots suitable to the execution window. To reduce the processing effort, in this case for the resource set one free time slot has to be selected according to an internal rating. If inserting a resource reservation, it is beneficial to select the earliest free time slot since the maximum difference between the planned finish-time and requested deadline provides most opportunities to handle and absorb resource failures. Consequently, the PoF should be the lowest for the earliest possible execution interval and any other time slots would not result in a lower PoF of the job.

Example 7.2.1
Figure 7.5 depicts an example schedule in which a reservation for a new job should be made. The new job runs 2 hours on 2 compute nodes. The earliest start time is 1:30 pm and its deadline is 11:30 pm. The assumption is made that because of the homogeneity of the compute nodes (see Section 6.1.1), any resource $Ri|i \in \{0, \ldots, 5\}$ conform to the SLA requirements and the job can run on any of these.

Possible reservations slots for the job execution are:

1. $E_{1|\{01:30,04:30\}} = \{R4 \text{ and } R5\}$
2. $E_{2|\{04:00,06:20\}} = \{R0 \text{ and } R1\}$
3. $E_{3|\{09:00,11:30\}} = \{R4 \text{ and } R5\}$

Since the resources in both sets $E_{1|\{01:30,04:30\}}$ and $E_{3|\{09:00,11:30\}}$ are the same, the second execution window on the resources $R4$ and $R5$ will not be considered in the remaining reservation process. Consequently, $R = \left\{E_{1|\{01:30,04:30\}}, E_{2|\{09:00,11:30\}}\right\}$.

Pre-selecting suitable resources is the first step of the risk-focusing reservation process. It only selects groups of resources fulfilling the requirements of the SLA concerning hardware and software constraints as well as time constraints. The only SLA parameter whose validity cannot be checked during this first phase, is the fulfillment of the maximum PoF defined by the consumer or adjusted by negotiation module. According to the complete reservation workflow (depicted in Figure 7.6) this feasibility check is performed directly after the resource pre-selection in the risk awareness and risk reduction phase. If not enough suitable resources could be pre-selected and the set R is empty, the scheduler can reject the SLA request since even without taking into account the PoF, the job execution is not feasible.

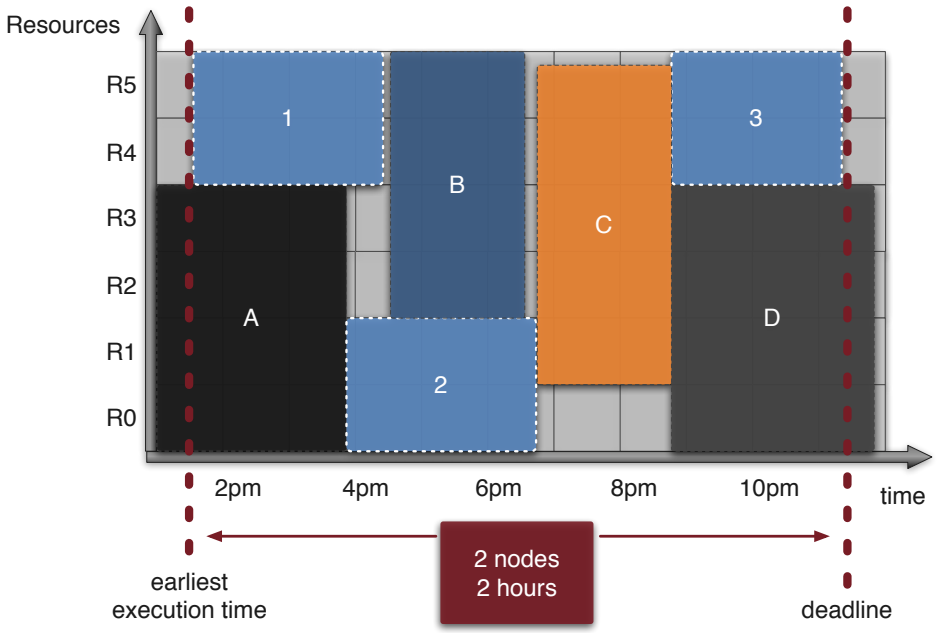


Figure 7.5: Example Pre-Selection of Resources in Schedule

7.2.2 Risk Awareness and Risk Reduction

The resource pre-selection in the first step is independent from the new approach of risk awareness and ensures through the consideration of job requirements the classical SLA provisioning. The next phase in the reservation process implementing risk awareness is crucial in order to estimate the risk of accepting the SLA and to compare the feasible PoF with the maximum value defined in the SLA request.

The risk awareness and risk reduction phase starts with assessing the PoFs of an SLA violation of j when using a reservation $E_{i|\{s,e\}}$, for $0 < i \leq |R|$. The risk assessment is initiated by the scheduler since making the reservation decisively depend on the PoF estimation. The reservation process implements thereby a targeted Risk Management process, as defined in Section 5.3. The assessment is out of the scheduler's scope and realised by a separate module which reports the estimated values within a list. To simplify the further process, this list has to be ordered either by the risk assessment module or the scheduler itself according to the PoFs of the resource sets.

The PoF ranking is the basis for resuming the reservation process. As described in Section 7.1 the negotiation module invokes a reservation process by determining an upper bound for the acceptable PoF which may differ from the contractor's requirement. If the maximum PoF condition can not be kept by any reservation, the scheduler would have to immediately reject the SLA request if no risk reduction is applied. Planning FT-mechanisms in scope of risk reduction enables to accept SLAs whose maximum accepted PoF could not be provided otherwise. In addition to the PoF estimations also customer policies could determine whether and which FT-mechanisms should be planned or considered precautionary. If at least one reservation conforms to the maximum PoF without applying risk reduction, all reservations having a

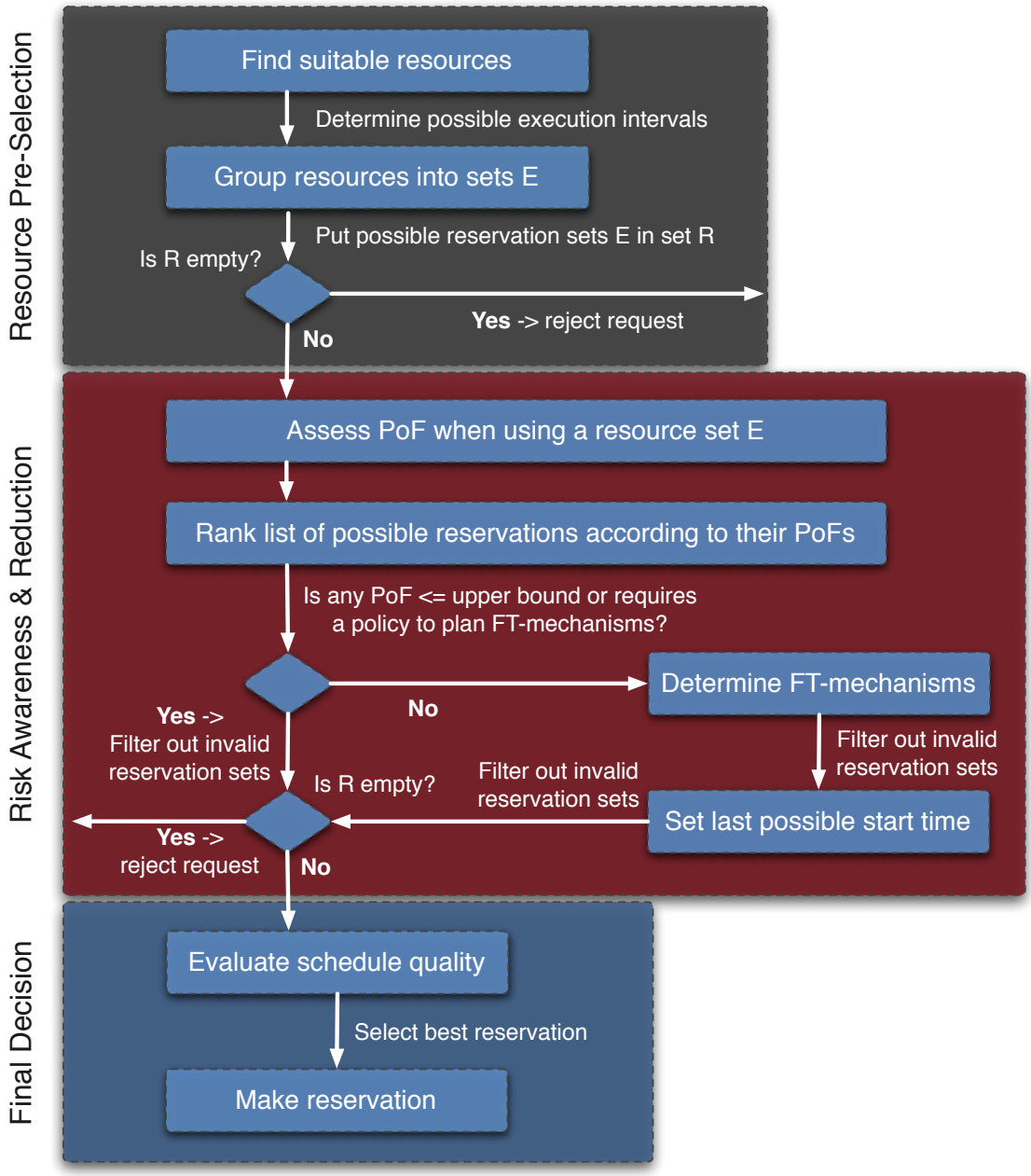


Figure 7.6: Risk Aware Reservation Process

higher PoF than the maximum value are filtered out of the set R. If no FT-mechanisms have to be planned because of policies, the scheduler proceeds with the final decision of selecting a reservation (see Section 7.2.3).

Using a planning based scheduler the consideration of FT-mechanisms in the negotiation phase is essential since performing these costs time: the reservation process has to extend the duration of the job execution if checkpointing should be performed (details are given in Section 7.4). Furthermore the buffer time for executing FT-mechanisms,i. e. the buffer is in

Figure 7.7 the latest completion time - t_p, depends on the planned start time, duration, and deadline; and thereby the selected execution window related to the possible time frame of the job is important. To perform FT-mechanisms, for example, in the case of migration, significant more time is spent than only generating a checkpoint since the data transfer has to be performed and the application resumes its execution from the latest checkpoint. In scope of risk awareness, comparing the buffer size between the planned end-time t_p and the deadline with expectations for the time needed to perform a migration is beneficial. This ratio should provide an indication of the successful execution of a migration according to the existing time constraints. If adequate statistics about network monitoring data are available, the risk assessment module can determine such an expected time. Note that this expectation is only meaningful if the standard deviation is low and the expected time is an accurate reference value. For external migrations, the expected time needed for the data transfer could be also determined if SLAs are agreed with network providers guaranteeing specific interconnection speeds to other providers. Dependent on the SLA, a challenge might be to pre-estimate the data volume of a checkpoint which has to be transferred if a priori only the input size is defined. In most cases the service consumer however defines the filesystem size used by the application within a SDT. This upper bound can be used to determine the expected time for a migration since the additional memory for storing register and network information within the checkpoint is usually small. The estimation of such values is out of scope of this work and might be considered within an enhanced risk assessment. Consequently, the success of a migration according to the buffer is reflected in the PoF when using specific resources in a determined execution slot. If according to the model the execution slot may be modified in order to execute other deadline jobs, whose requests have been inserted later, not considering the buffer ensures that PoFs, which have been estimated during the SLA negotiation, are not modified because of a replanning.

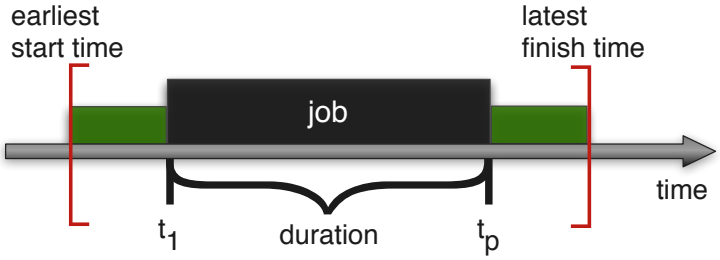


Figure 7.7: Buffer for Performing FT-mechanisms equals Deadline - t_p

Dependent on the applicable FT-mechanisms, different activities can be planned in scope of risk reduction. The decision which FT-mechanism is planned has to take into account the probability of not finding enough suitable free alternative resources for resuming/restarting the job when resources have failed. It is obvious that performing FT-mechanisms consumes time and resources and consequently applying them results in additional cost. Consequently, the provider follows the objective to initiate that FT-mechanisms resulting in fulfilling the upper bound with the lowest costs. To select the less expensive one, these are tested as shown in Listing 7.1.

Listing 7.1: Testing to apply which FT-Mechanism

```
1  p := feasible PoF;
2  m := max PoF;
3  Z := list of applicable risk reduction plans; ordered by increasing costs
4
5  if (p > m) {
6   for(z is head of Z; z has more elements; next element) {
7     p' := p;
8     c  := P(a alternatives available | a alternatives needed);
9     plan z;
10    p' := feasible PoF | c is valid and z is performed;
11    if (p' ≥ m) {
12       p := p';
13       break;
14    }
15  }
16 }
```

Since performing FT is time-consuming, the scheduler estimates for each possible reservation E a *Latest Possible Start-time* (lps). Note that such considerations are not necessary if any of the reservations fulfill the PoF requested. This would be only make sense, if different reservations implies different internal cost. In this case the provider could save money if running job j on the resource set E_1 instead of resource set E_2 even if additional FT-mechanisms have to be performed and paid. Since the model assumes a homogeneous cluster, the costs for using a compute node should be the same for all and consequently, no differentiations have to be made in the reservation process. Determining the lps depends on the plan z which would be applied and has to be performed in the case that the new feasible PoF p' is not higher than the maximum m. Accordingly, setting the lps should be done in the then-clause after line 12 in Listing 7.1.

The lps definition depends primarily on time constraints of the SLA, i. e. deadline and execution duration. The (expected) time for performing the FT-mechanism is also crucial. Instead of actual times, often only expected times can be considered since these have to be estimated and cannot be defined precisely a priori. For example, the time needed to perform a migration significantly depend on the checkpointing size. This has to be estimated a priori during the SLA negotiation. Considering the size is very important because in the case of failure it has to be transferred to another compute node. Especially, a transfer to a remote site is critical since the interconnection of different sites will never be as fast as the internal network (see Section 6.1.1). That is why the resource utilisation has to be also considered in the lps definition in order to assess whether local resources will suffice to perform a migration or job restart. Consequently, the lps definition has to consider time constraints of the SLA and also FT-actions z planned. By defining $\text{lps}_{j|z}$, the lps depends on the required time for an FT-action z which is planned according to aspects of job j: deadline, input data, and planned action (which depends on the availability of alternative resources).

Note that additional time could be considered for performing fault-tolerance but no explicit FT-mechanism is planned in scope of risk reduction. The PoFs of the resource reservation are the decisive factor whether to add time for one (and which) FT-mechanism. The number of available alternative resources will also influence the type of a fault-tolerance mechanism (reserving additional spare resources, execute the job redundantly). In addition, customer

policies could require specific FT-mechanisms to be planned. Since these policies might depend on the ratio of feasible PoF and the maximum accepted upper bound, the process checks the initialisation of FT-mechanisms after the PoF estimation. If policies define to plan FT-mechanisms in either case, i. e. PoF independent, these can be considered in the first step of this workflow – during the pre-selection of resource.

After determining the FT-mechanisms which should be taken into account by a time buffer, the scheduler continues filtering according to the lps. Since the lps depends on the assessed PoF for a resource set E, the lps for resource sets with different PoFs may vary. The filter process compares the time slots according to the job's duration and to the PoF dependent lps, i. e. time slot.end $-$ duration $\geq$ lps. As mentioned above, for single jobs running on one compute node the earliest time slot is selected if a resource has more than one possible slot for the job execution. That is why resources whose time slot.end $-$ duration $\geq$ lps, have no useable time slot at all. Furthermore selecting the first time slot implies that, if a resource usage results in a too high PoF, no other time slot will have a lower PoF since the time available for performing fault-tolerance is lower and this time should influence the PoF. Hence, it ensures that the filter process does not leave a possible execution slot out of consideration. It is important to note that the definition of the lps takes into account that an FT-mechanism can be performed. However, it does not result in an extension of the job duration. The duration should only be extended if an FT-mechanism is initiated which requires time on the same resource, such as checkpointing. To check the feasibility of the initiation of an FT-mechanism, only applicable risk reduction plans are considered for each reservation set (see line 3 of Listing 7.1). Here, the applicability is checked in relation to the supported FT-mechanisms in general as well as to the reservation set E.

After ensuring by the resource pre-selection phase that the classical SLA requirements are fulfilled, the upper bound for the PoF is validated in the risk awareness and risk reduction phase. The PoF is estimated for the job j when using a possible reservation set E. If the upper bound cannot be fulfilled by any reservation set, the risk reduction is initiated. In this scope the applicability of FT-mechanisms are evaluated and, if necessary, an FT-mechanism is precautionary planned. The scheduler follows the strategy to add for each set E the cheapest FT-mechanism, which achieves the maximum accepted PoF, in order to reduce the profit as less as possible. Reservation sets E which cannot fulfil the maximum PoF even if FT-mechanisms are planned, are filtered out the set R. If the risk awareness and risk reduction phase results in an empty set R, the scheduler has to reject the SLA request. Note that the provider can also define constraints which FT-mechanisms are allowed to be planned in scope of risk reduction. Allowing to plan a limited set of possible FT-mechanisms is meaningful in order to make no reservations being beyond the profit margin.

7.2.3 Final Decision and Make Reservation

If the set R contains at least one resource set E at the beginning of the final decision phase, the scheduler can make an advance reservation for the job request. This implies that resources are available which fulfil the hardware, software, time as well as PoF constraints of the SLA. In scope of risk reduction, FT-mechanisms could be taken into account by determining the latest possible start time or planning those. If no risk reduction plans have to be applied since the

upper bound of the PoF could be achieved, the lps conforms to the deadline − job.duration. The resource pre-selection has generated resource sets which obtain a free and possible larger time slot for the execution. Usually the free time slots on eligible resources are larger and the planned start time is not completely fixed after the pre-selection phase. As a default approach, the job execution could start at the beginning of the reservation slot. However, the schedule quality could be badly influenced if the job start is planned always on its earliest start point which might be equal to the beginning of the reservation. For instance, in Example 7.2.1 the resources $R4$ and $R5$ are free at least half an hour before the earliest execution time of the job, i. e. 1:30pm. Hence, this gap is perhaps not usable by any other job. Since in addition to such schedule aspects the lps has to be considered, the final decision to of making a reservation includes the definition of the planned start time.

The SLA defines the job submission time which conform to the earliest start time, execution duration, and deadline. The lps has been assessed according to the deadline, to the probability of failure of the SLA when using a specific resource set E, and, if necessary, to the applicability of FT-mechanisms. Searching adequate starting points $s \in \{r_s, r_e\}$ for a possible reservation $E_{i|\{r_s,r_e\}}$ is restricted to these values:

- $s \geq$ `job.earliestExeTime`
- $s+$ `duration` $\leq$ `job.deadline`
- $s+$ `duration` $\leq r_e$
- $s \leq \text{lps}_E$

Let the determined time slot from resources $r \in E_{i|\{r_s,r_e\}}$ ranges from t_1 to t_n. The validity of $t_n - t_1 \leq$ job.duration is mandatory. If in scope of risk reduction, FT-mechanisms have been planned which are time-consuming, an extended job duration is planned instead of the duration specified in the SLA. Hence, the constraints above use the variable `duration` instead of `job.duration`.

In the next step of the scheduler's reservation workflow the starting point in $[t_1; t_n]$ has to be identified which achieves the highest schedule quality. In most cases and if possible, a start time at the beginning of the time slot will be advantageous. For determining the most advantageous starting point a comparing measurement is required which has to estimate the effects of a job execution on the whole schedule. For example if a job will completely fill a single fragment, then the schedule is optimal for that segment. Accordingly, a measurement of benefit will maximise and one of detriment will minimise. However in most cases a reservation will degrade the schedule's quality because new fragments result if the reservation cannot be placed immediately after or before another job execution. Section 7.2.4 presents a useable measurement for selecting the best starting point.

This internal comparison leads to the definition of a specific starting point s for each reservation $E_{i|\{s,e\}}$ which is optimal for the schedule's quality. In the next step the effects on the schedules have to be compared. Finally, the reservation should use the resources with the best/less worse effect on their local schedule. It is important to note that not the resource set should be selected whose resultant schedule has the best quality over all. The change of the quality is essential because otherwise first of all schedules of resources having a good quality would be further improved. Section 7.2.4 describes a useable measurement for comparing different quality changes. After comparing the schedule's quality changes of admissible resources, performing

the job execution on resources of one reservation $E_{i|\{s,e\}}$ has been identified as the most beneficial for balancing PoF and schedule quality change. If the best reservation candidates concerning the quality change have similar values, the PoF should be taken into account when selecting one reservation set. This means that if two reservation sets E_1 and E_2 have a similar change in quality and no other reservation set is better from R, then the decision should depend on the PoF when using E_1 or using E_2. The repeated consideration of the PoF results in a good balance of schedule quality and PoF. After selecting a reservation set, the scheduler determines the associated resources, makes an advance reservation for the job execution, and notifies the negotiation module. The negotiation module is also notified if no advance reservation according to the requirements could be made. In this case it would be beneficial to forward additional information whether either no resources has been free or the failure results from the risk awareness, i. e. the estimated PoF is higher than the upper bound and no adequate FT-mechanisms could be planned to keep this constraint.

An idea to control the reservation making in scope of the PoF is that the negotiation module determines a strategy of the PoF. This means that it can commission the scheduler to make an advance reservation having the lowest PoF or having a PoF which is close by the maximum value. However the consideration of an additional aspect results in a reduction of the importance of the schedule quality.

The final decision phase makes an advance reservation if candidates exit in the set R. The resource pre-selection and risk reduction phase have ensured the conformity of the resource sets E_i to the SLA. Consequently, these aspects can be left out of consideration in the final phase. Before determining a resource set $E \in R$ which should be assigned to the job, the explicit planned start time of the job is defined and the impact of this reservation on the schedule is considered. Measurements which can be used for comparing different starting points within a reservable time slot as well as for the schedule quality are presented in the following section. If the estimated values for reservation sets E_i with the best change in the schedule quality are similar, the final selection process depends on the PoF estimation.

7.2.4 Example Measurements for Risk-Focusing Reserving

The final decision process evaluates for each resource set E_i the change of the schedule quality if this reservation would be made for an SLA bound job j. Decisive for the schedule quality change is the planned start time of the job, since in the resource pre-selection phase free execution slots have been rated as a candidate slot if the time slot length is equal *or longer* than the job duration. The candidate selection used the simplification that if multiple execution slots on the same resource set is possible, the earliest one is added to the set R. Section 7.2.4.1 clarifies this aspect and the requirements to be considered when defining a measurement for the candidate selection.

To compare different reservation candidates $E_i \in R$, measurements for the starting point and the schedule quality modification have to be defined. In this section two exemplarily measurements are presented: first of all Section 7.2.4.2 presents an approach to determine the best starting point of a time slot. Afterwards in Section 7.2.4.3 measuring the change of the schedule quality is presented. A final example is shown in Section 7.2.4.4.

7.2.4.1 Select Execution Slot as Candidate for R

Section 7.2.1 followed the assumption, that only the earliest possible execution slot on the same resources is added to the set R. As a consequence, in Example 7.2.1 on the resources $R4$ and $R5$ the job could be executed either between 01:30 pm and 04:30 pm or between 09:00 pm and 11:30 pm. Since only the first possible execution slot is added to R, $E_{1|\{01:30,04:30\}} \in R$ and $E_{3|\{09:00,11:30\}} \notin R$. This simplification is admissible since, if the resource provider makes good offers, the workload of its resources will be high and consequently not many free execution slots will be available. In the case that a measurement should be used to select one of all possible execution slots on the same resources, the lps has to be taken into account. Since however, the PoF is not known during the resource pre-selection, the latest start time has to conform to the earliest possible lps which could be defined according to the job's time constraints and risk reduction plans. This means

$$\text{start} \;\leq\; \text{j.deadline} \;-(\;\text{j.duration} \;+\max\{\text{FT-time}\})\,. \tag{7.1}$$

7.2.4.2 Find the Best Starting Point

Let the determined time slot on resource set E ranges from r_s to r_e and let j be the job to be scheduled. This includes that on all resources $r \in E$ the previous job is planned to end at the latest at time $r_s - 1$ and the next job is planned to begin at the earliest at time $r_e + 1$. However, the schedule for resources $r_i \in E_i$ might differ: the planned job completion time before r_s can be $r_s - x$ for resource $r_i \in E$, and $r_s - y$ for resource $r_j \in E$ with $x \neq y$. For non-parallel jobs[1] the best starting point can be defined which is the most promising solution for the resource to be used. Jobs running on several compute nodes in parallel have to balance the effects of the starting point on all resources involved. Consequently, the starting point might be not the optimal choice for all resources, but on average or for the majority of resources it is the best one.

In the following, *j.earliestExeTime* is the negotiated time at which the job data will be at the latest available in the RMS. Analogously, other negotiated values of the SLA are referred with j.$*$. A starting point $s \in [r_s, \min\{\text{lps}_E, (r_e - \text{j.duration})\}]$ is searched. Accordingly the starting point depends not only on the free time slot, it cannot be started after lps_E in order to not violate any SLA constraint.

To simplify the reading of the developed measurement, variables are defined to avoid the maximum statements. *Estart* is defined as the earliest starting point of the job in $[r_s, r_e]$ with *Estart*:= r_s since in the definition of r_s the job's earliest possible start time was considered. *Lstart* is the latest starting point of the job in $[r_s, r_e]$ and equals lps_E since the lps has been determined in relation to r_e and j.duration

An optimal positioning of the starting point results in that no additional fragment is generated. Hence, the selection process checks first of all whether the job can be positioned in that way that the number of fragments does not increment. Obviously, for each resource schedule the number of fragments should be reduced as much as possible since this enables to execute other jobs with any duration $\leq$ *slot.length*. In counting fragments an ϵ_f-tolerance should be

[1] using only one compute node

taken into account, i. e. slots $[t_h, t_l]$ with $\delta t = t_l - t_h \leq \epsilon_f$ are negligible and not counted as a fragment. The selection process has to determine the number of fragments for each resource $r_i \in E$ by considering δt between the job execution and the previous job p and the succeeding job s (see Figure 7.8). For each r_i the number of resulting fragments is zero if the reservation would completely fill the gap between two jobs. The number is 1 if the job would be executed directly after or directly before another reservation, i. e. after p or before s. In each other case the number equals to two, which implies that both δt are bigger than ϵ_f. To be less strict, the sum of time between the planned and the existing reservations can be compared to $2\epsilon_f$ and if the sum is not higher, the resulting fragments are assumed to be one instead of two.

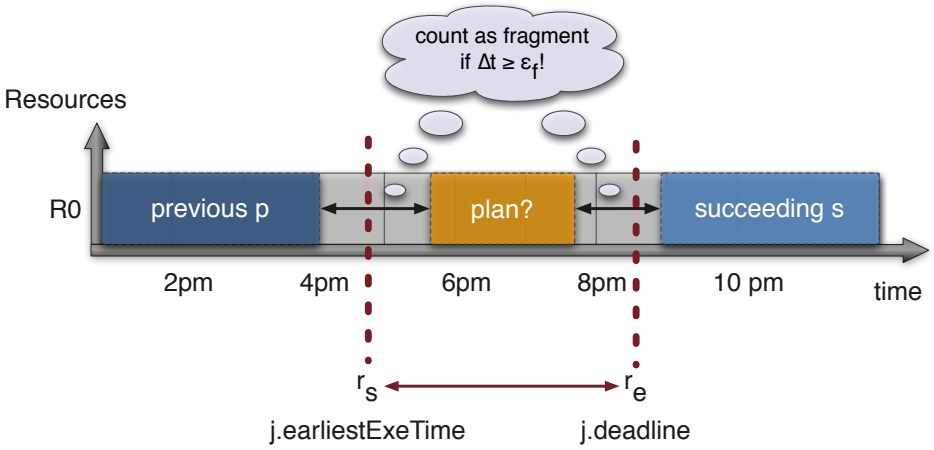


Figure 7.8: Consider Time Differences when Counting Resultant Fragments

The description above considers the counting of fragments for one resource. Since job j uses n resources, the total number of fragments is in the interval $[0; 2]$ for each resource r_i and for all resources in $[0; 2n]$. If the sum of resulting fragments is less than n, on average the number of fragments does not increment when making the reservation which implies a good starting point. If however, the sum is between n and $2n$, further comparisons may be made in order to determine a good starting point. The following algorithm is defined:

Listing 7.2: Start-Point Determination

```
1  n := number of nodes requested by job j;
2  e := determineNumberOfFragments(Estart −εf);
3  l := determineNumberOfFragments(Lstart +εf);
4
5  if (e ≤ n)
6     s := Estart;
7  elseif  (l ≤ n)
8     s := Lstart;
9  else
10    make further comparisons;
```

Checking the number of resultant fragments can be omitted for a better performance if *j.earliestExeTime* is significantly larger than the time of the SLA negotiation. In this case it can be suggested that at the evaluation time not many reservations for the execution slot

are yet planned and accordingly positioning the job immediately before or after another job is very improbably.

If job j cannot be positioned in that way that the number of fragments does not increment, the number of average jobs executable around the reservation should be considered. Thus, the measurement works with the average duration of jobs, set here as μ. In an more enhanced concept, jobs could be classified in short, middle, and long-time jobs and by considering different μ_i, a more precise comparison of different starting points is possible. Here, only one example measurement is shown in order to point out the idea of comparing different starting points. Consequently, only one μ is considered and the assumption is made that jobs vary little in their duration. The value of μ has not strictly be the average duration. If, for example, 90% of all jobs have an duration less than ten minutes, it could be advantageous to set $\mu = 10$ minutes. The definition of μ is system- and customer-specific and has to be set from administrators.

To determine the starting point, the number of full average jobs which can be executed before and after the execution of j starting at point s is estimated for each resource $r_i \in R$, denoted as k_i. A tolerance of σ, the standard deviation of μ, is added since μ is an average. On resource r_i the previous job p before r_s completes at time t_b and the succeeding job s is planned to be started at time t_a which might be equivalent to r_s and r_e. As possible starting points, $s =$ Estart, $s =$ middle of possible starting points, and $s =$ Lstart are considered. In most cases calculating $s =$ middle of possible starting points can be omitted since it will often not be a good position. However, for the completeness it is also mentioned here. To speed up the calculation, in general only values for $s =$ Estart and $s =$ Lstart can be assessed.

1. $$k_i(\text{begin}) := \left\lfloor \frac{\text{Estart}-t_b}{\mu} + \sigma \right\rfloor + \left\lfloor \frac{t_a-(\text{Estart}+\text{j.duration})}{\mu} + \sigma \right\rfloor \tag{7.2}$$

2. $$k_i(\text{end}) := \left\lfloor \frac{\text{Lstart}-t_b}{\mu} + \sigma \right\rfloor + \left\lfloor \frac{t_a-(\text{Lstart}+\text{j.duration})}{\mu} + \sigma \right\rfloor \tag{7.3}$$

3. $$\text{middle} := \text{round}\left(\text{Estart} + \left(\frac{\text{Lstart}-\text{Estart}}{2}\right)\right)$$
$$k_i(\text{begin}) := \left\lfloor \frac{\text{middle}-t_b}{\mu} + \sigma \right\rfloor + \left\lfloor \frac{t_a-(\text{middle}+\text{j.duration})}{\mu} + \sigma \right\rfloor \tag{7.4}$$

Note that it is important to only assess the **whole-numbered** executable average jobs since otherwise the number k_i is the same for all starting points. The starting point with the highest $\sum_i k_i$ should be selected. Accordingly, this measurement should be maximised. Often k_i(begin) and k_i(end) will equal. In these cases the early beginning is selected in order to obtain opportunities for later jobs and to have a bigger buffer for performing fault-tolerance for job j. After selecting the position of the starting point $s \in [r_s, r_e]$, s can be shifted σ time units. This fine tuning can be realised by a trivial algorithm for non-parallel running jobs: it has to find an ϵ with $-\sigma \cdot \mu \leq \epsilon \leq \sigma \cdot \mu$ so that $s + \epsilon \mod \mu = 0$ and Estart $\leq s \leq$ Lstart. For parallel jobs using several compute nodes at the same time, the fine-tuning can be performed in that way that for the maximum number of resources $r_i \in E$ $s + \epsilon \mod \mu = 0$ is valid.

If j suits well in $[r_s, r_e]$ on each resource, on average not more than one fragment results from positioning the reservation in the resource's schedule. In this case the total number of fragments is $\leq n$ and the calculations of k_i would not be initialised. Since the number

of resulting fragments are important for the positioning of later arriving jobs, this indicator should be used in the estimation of the modified schedule quality as described in the following section.

7.2.4.3 Estimate the Schedule's Quality Change

The schedule's quality depends on the position of the determined starting point $s \in [r_s, r_e]$ as well as on the number of resulting fragments. On resource r_i the previous job before r_s completes at time t_b and the succeeding job is planned to be started at time t_a which need not be equivalent to r_s and r_e. The criteria for the schedule's quality are the number of resulting fragments and the probably unusable time $u_{r_i}(s)$ on resource r_i. Probably unusable time is defined as the time which cannot be used by an averaged job. Accordingly it is calculated by

$$u_{r_i(s)} := ((s - t_b) \bmod \mu) + ((t_a - (s + \text{duration})) \bmod \mu) \tag{7.5}$$

This unusable time has to be compared with the unusable time without executing the job $\left[u_{r_i(\neg s)} := ((t_a - t_b) \bmod \mu)\right]$. The resulting ratio

$$q_{r_i(s)} := \frac{u_{r_i(s)}}{u_{r_i(\neg s)}} \tag{7.6}$$

can be used to measure the schedule's quality change. The job execution leads to an improvement of the quality if $q_{r_i(s)} < 1$. Furthermore, the changed quality is the worse the higher $q_{r_i(s)}$. The sum of unusable time and the ratio should be minimised for all resources $r_i \in E$.

The other criteria is the number of resulting fragments and can only reach for each resource r_i the values 0, 1, and 2. It is simply computable, accordingly, the process is not mentioned here. Obviously, the number of fragments should be minimised. Beyond the number, especially the size of the fragments is important. If fragments are larger than the average job duration, the fragments are probably still useable. However, little fragments will mostly not be used and accordingly the quality of the schedule reduces. A function whose estimated value will modify the $q_{r_i(s)}$ was developed. The calculation of the function f is only important for those fragments with length $x \leq (\mu - \sigma)$ where σ is the standard deviation of the average job duration. Accordingly, $f(x)$ is zero for all other values of x. The function f is defined as:

$$f(x) = \mu \cdot \left(e^{\frac{2x}{\mu - \sigma}}\right)^{-1}, \text{ if } x > (\mu - \sigma)$$

By using function f it is achieved that the smaller the resulting fragments the worse the quality. Namely the function plunges down from μ to 0 by considering also σ. Figure 7.9 shows an example curve of $f(x)$ for $\mu = 30$ and $\sigma = 5$.

The overall estimation has to combine both criteria to determine the change in quality. The resources should be ordered first of all rising in the total number of fragments. The second order will result from the quality of unusable time and size of fragments:

$$Q = \sum_i q_{r_i(s)} = \sum_i \left(\frac{u_{r_i(s)}}{u_{r_i(\neg s)}} + \frac{f(i) + f(j)}{\mu} \right) \tag{7.7}$$

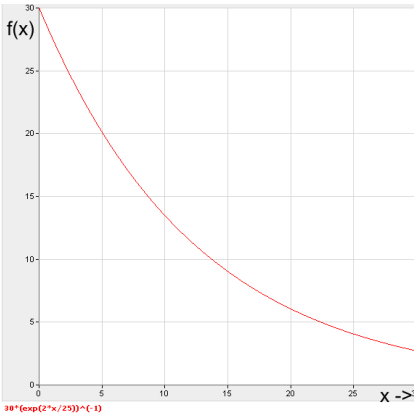


Figure 7.9: Reduced Schedule Quality because of Small Fragments

This modified measurement $q_{r_i(s)}$ realises that the fragment size is appropriated weighted according to the unusable time since the new term is a ratio and no absolute value. The modification does not influence that the higher the values the worse the quality change. If several resource sets E have the same values of Q within a tolerance of $\epsilon > 0$, the PoF is the decisive criteria when selecting one resource set for the reservation.

7.2.4.4 Example Computation

This section presents an example of using the suggested measurements of Section 7.2.4.2 for defining the exact starting point and Section 7.2.4.3 for comparing schedule qualities. For simplification the example considers the scheduling of a single, non-parallel job and R contains only two suitable resources conforming to the SLA requirements. The job j has the earliest execution time 12, needs 4 time units, and its deadline is defined as 21. In the considered system the average duration of a job is 4 time units. All jobs have nearly the same execution duration so that the standard deviation of the average job duration is $\sigma = 0,3$.

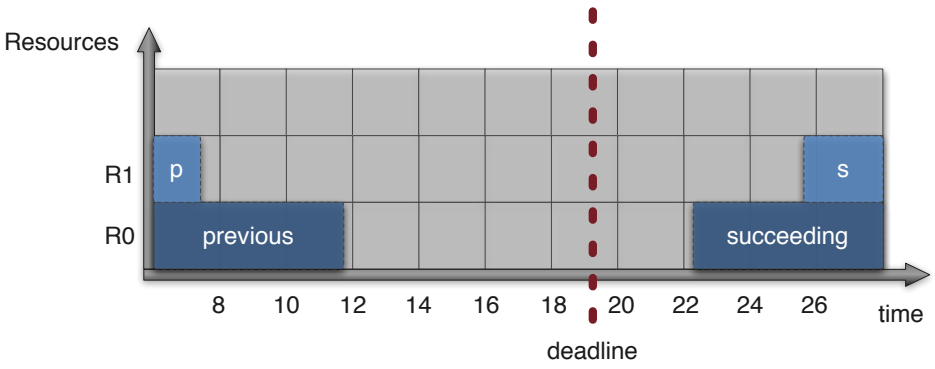


Figure 7.10: Example of Schedules of two Suitable Resources

Resources R_0 and R_1 suit to the job requirements. The job j can be executed on resource R_0 in the time slot $[11, 22)$ and on R_1 in $[7, 24)$ (see Figure 7.10). The right side of the interval

is not including which means that the job has to stop just before 22 on R_0 and 24 on R_1. Let the PoF of an SLA violation executing the job on R_0 be 9% and on R_1 5%. Those values are both below the upper bound of 10% asked by the SLA contractor; consequently no additional FT-mechanisms have been planned.

Since the job data is submitted at 12 and both time slots start earlier, the values of *Estart* equal of both resources: $\text{Estart}(R_0) = \text{Estart}(R_1) = 12$. The *Lstart* depends on the *lps* and on the deadline. Since both resources have low enough PoFs, the latest starting point conforms to $\text{lps}_{R_0|R_1} = \text{Lstart}(R_0|R_1) = 21 - 4 = 17$. The job execution on both resources produces an *additional* fragment in their schedules, this means that in both schedules two fragments exist if placing the reservation in it - one before the job and one after its job execution.

First of all the best starting point is calculated. The resulting estimations for resource R_0 are:

$$k(\text{begin}) = \left\lfloor \frac{12-11}{4} + 0,3 \right\rfloor + \left\lfloor \frac{22-(12+4)}{4} + 0,3 \right\rfloor = 1 \tag{7.8}$$

$$k(\text{end}) = \left\lfloor \frac{17-11}{4} + 0,3 \right\rfloor + \left\lfloor \frac{22-(17+4)}{4} + 0,3 \right\rfloor = 2 \tag{7.9}$$

$$k(\text{middle}) = \left\lfloor \frac{15-11}{4} + 0,3 \right\rfloor + \left\lfloor \frac{22-(15+4)}{4} + 0,3 \right\rfloor = 2 \tag{7.10}$$

Following, the job should start at 17 since the end position is favoured if the value is equal to the $k-$value of the middle start position. Calculations for resource B:

$$k(\text{begin}) = k(\text{middle}) = k(\text{end}) = 3 \tag{7.11}$$

In that scenario all position have the same k-value. The start position is set at the beginning in order to remain free time slots for later arriving job requests. The next step in the exact definition of the start point is the fine tuning. Therein, the start point can be shifted $\mu \cdot \sigma$ time units into both directions. However, in these schedules a fine tuning has no benefit and the start points are kept. The next step is the comparison of the schedule qualities according to the measurement of Section 7.2.4.3.

Resource R_0:

$$u_{R_0}(17) = ((17-11) \bmod \mu) + ((22-21) \bmod \mu) = 3 \tag{7.12}$$

$$u_{R_0}(\neg 17) = ((22-11) \bmod \mu) = 3 \tag{7.13}$$

The two fragments $[11, 17]$ and $[21, 22]$ have the size of 6 and 1 time units. $f(6) = 0$ because $6 \geq \mu - \sigma_\mu$. Since $f(1) \approx 3.064$, $\frac{f(1)}{\mu} \approx 0.766$. Following:

$$q_{R_0}(17) = \frac{3}{3} + 0.766 = 0.766 \tag{7.14}$$

Thus, the quality of R_0's schedule is a little bit better with than without executing the job.

Resource R_1:

$$u_{R_1}(12) = ((12-7) \bmod \mu) + ((24-16) \bmod \mu) = 1 \tag{7.15}$$

$$u_{R_1}(\neg 12) = ((24-7) \bmod \mu) = 3 \tag{7.16}$$

The two fragments $[7, 12)$ and $[16, 24)$ have the size of 5 and 8 time units. Both fragments have a length over the average value μ and following the function values $f(5)$ and $f(8)$ are zero. Therefore:

$$q_{R_1}(12) = \frac{1}{3} = 0.33 \tag{7.17}$$

Running the job on resource R_1 will significantly improve B's schedule quality. Comparing with the effect on resource R_0, the resource should be scheduled on resource R_1. Please note that the execution on R_0 would have a higher PoF which however fulfils the upper bound. If no schedule qualities would be considered, $R0$ is preferred since it fulfils the maximum accepted value and R_1 is still useable for other jobs requesting a lower PoF. However, since both PoFs are acceptable according to the negotiation policy and the prime objective is the schedule quality, the selection is valid.

The usage of example measurements were demonstrated in this section. Providers may define different measurements for selecting the starting point and determine which resource set $E \in R$ should be used for a reservation. The presented risk-focusing reservation process can also use simple heuristics for both decisions. The main aspect of the risk-focusing workflow for making reservations is that first of all several candidates are selected, afterwards the PoF is evaluated and in the last step the schedule quality is considered. If the feasible PoF is higher than the upper bound, risk reduction is essential in order to achieve a contract conclusion of the SLA. However, the PoF is not the only objective of the reservation making since the schedule quality is important in order to build a basis for a good system utilisation. The consideration of this aspect has lead to that the final decision is made based on the schedule quality. However, if schedule qualities of different reservation sets are equal (or nearly similar) the PoF are again the decisive factor. In particular, if considering the PoF estimations for similar schedule qualities, a great balance of PoF and schedule quality is achieved. The described workflow forms the basis for a specific scheduling strategy considering PoFs and can be adjusted by defining arbitrary measurements.

7.3 Combining Risk Awareness with Arbitrary Scheduling Strategies

The scheduling process presented in Section 7.2 focuses on risk awareness combined with estimating the resulting schedule-quality when making a reservation. However, providers are often not willing to change their scheduling policies completely to this concept as the strategies used depend on specific policies or have been proven and established over time. As a consequence, it would be beneficial to couple existing scheduling strategies with Risk

Management activities. To avoid constraints concerning useable scheduling mechanisms and RMS, risk awareness has to act on a different layer than arbitrary scheduling policies.

Decoupling risk awareness from the core scheduling mechanisms results in decision process with multiple layers. First of all, the risk aware layer might define or change requirements of the job execution. For instance, the risk awareness demand for performing checkpointing for each SLA bound job. In this case, the risk aware layer modifies the job description by adding the checkpointing functionality. Customers might ask also explicitly for checkpointing in their SLA. If necessary, the risk aware layer might then increase this checkpointing frequency. The SLA request is then forwarded to the original scheduling process positioned one layer below the risk awareness (see Figure 7.11).

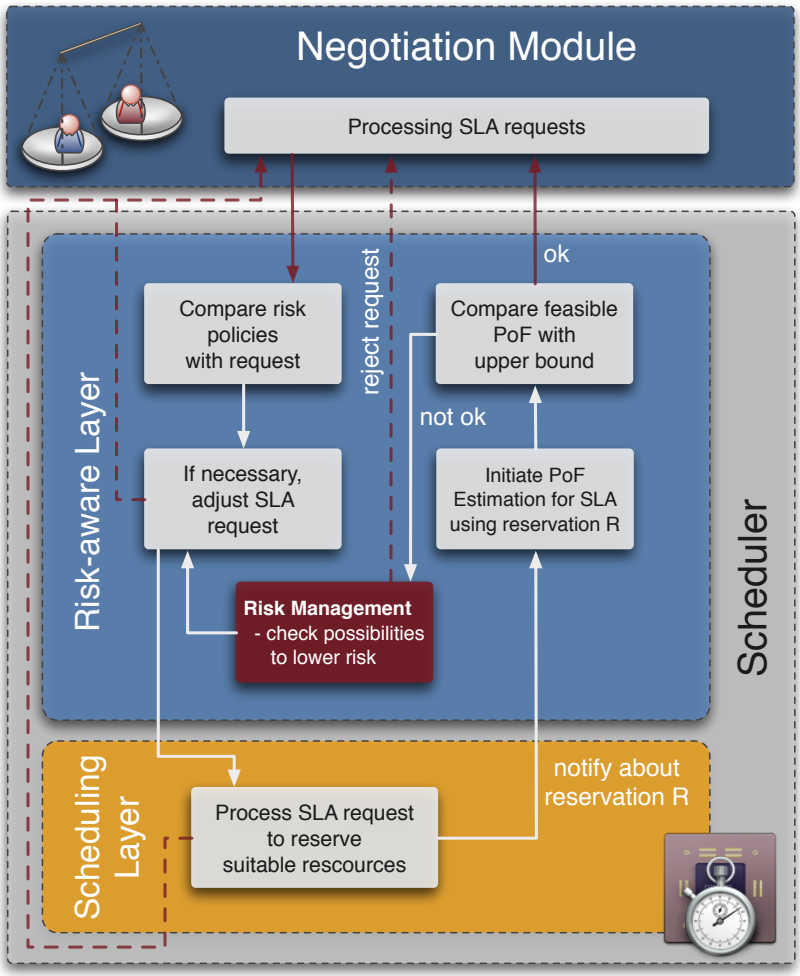


Figure 7.11: Risk Aware Reservation Process Coupled with Arbitrary Scheduling Strategies

The scheduling process is executed without any modification and selects the resources for the job execution according to its defined resource and customer policies. Afterwards, the possible reservation slot is published to the superior risk aware layer which initiates the estimation of the PoF. The estimated PoF is compared with the upper bound defined by the contractor

or which has been modified by the negotiation module. If the estimated PoF is lower than or equal to the upper bound, the RMS can accept the SLA request and the final decision is made by the negotiation module. If the estimated PoF is higher, a Risk Management evaluation has to be performed in order to determine whether in context of risk reduction functionalities can be added to provide the PoF required (see Section 7.4). In most cases this leads to an repeated evaluation of the scheduling mechanism. In either case the negotiation module is notified about the acceptance or rejection of the RMS and decides to agree or reject the SLA offer. Note that even if the estimated PoF is lower than the accepted bound, the negotiation module could reject the SLA offer because of internal or contractor policies. For instance, the negotiation module only accepts SLA bound jobs whose feasible PoF is higher than 10%. Instead of checking this after finding a suitable reservation, it is more efficient if the negotiation module performs such policy evaluation checks before engaging the scheduler to make an advance reservation. Adjusting the maximum accepted PoF in the SLA request is a simple mean that the reservation conforms to such policies.

The Risk Management activities integrated in the reservation process describe a targeted Risk Management process. After the risk aware layer has received the notification about the reserved resources, the risk aware layer initiates the PoF estimation. Consequently, the event of the notification triggers the PoF estimation and a monitoring of this event is performed in the risk aware layer of the scheduler (see Section 5.3). Since a separate module is responsible for the PoF estimation, the scheduler becomes again active in the decision making process of the targeted Risk Management process. Part of the decision making is the risk evaluation which is performed by comparing the estimated PoF with the maximum accepted value. If the estimated PoF is not higher than the upper bound, the scheduler accepts the risk along with the agreement of the SLA. The negotiation module determines the price according to the PoF and thereby the risk acceptance constraints can be defined by provider's policies. If the risk evaluation results in that the feasible PoF is too high, the decision process is continued in the block denoted with Risk Management in Figure 7.12. In this case the possible modifications of the SLA are evaluated in comparison with the effect and costs of the modifications. An adjustment belong to the Risk Management strategy risk reduction (see Definition 3.1.4) and would be performed as part of the risk treatment. In most cases a modification implies a repeated scheduling in order to evaluate the feasibility of the adjusted SLA requirements. Hence, the repeated scheduling process is also part of risk treatment.

To combine risk awareness with arbitrary scheduling strategies, decoupling Risk Management steps from the scheduling workflow is necessary. Consequently, a two layer approach strictly separates the scheduling process from the steps performed in scope of risk awareness. The risk aware layer delegates the scheduling layer to reserve resources according to standard mechanisms established in the provider's RMS. The risk awareness and Risk Management is inserted in a superior layer using the reservation which was made by the unmodified scheduler. The risk aware layer can modify the SLA according to risk policies before the reservation is made or can adjust requirements to lower the feasible PoF. The negotiation module is responsible for making the final decision about acceptance or rejection of an SLA. This will primarily be based on the feasible PoF but might also depend on contractor (specific) or internal policies.

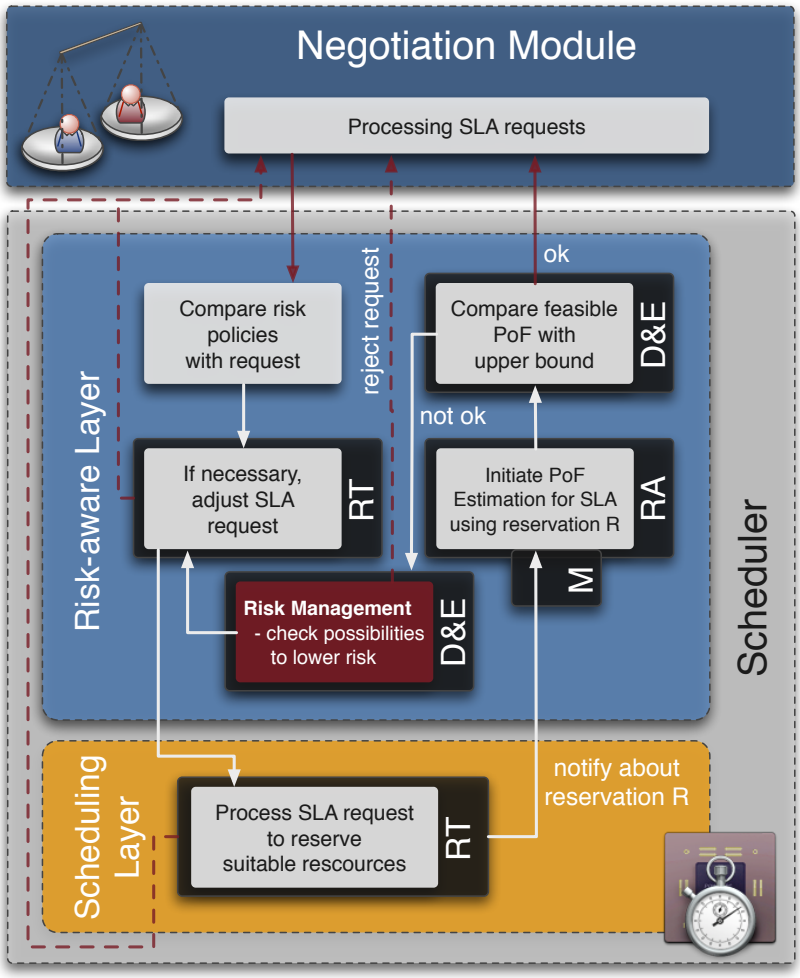


Figure 7.12: Highlighted Steps from Targeted Risk Management Process in Reservation Workflow

7.4 Risk Reduction during Negotiations

Risk awareness in the SLA negotiation and the underlying reservation process can lead to conflicts regarding the estimated PoF and the maximum value accepted from the contractor. In addition policies might request to internally lower the upper bound defined by the contractor. Whether the upper bound was set by the contractor or modified by the provider itself, must not be differed in the scheduling process since it conforms to the PoF the scheduler's reservation should fulfill. As a consequence of considering failure probabilities it is possible that an SLA bound job could not be accepted even though the scheduling process was able to make a reservation fulfilling resource requirements and time constraints defined in the SLA. In particular, if the difference between estimated and maximum PoF is low, it might be more profitable to lower the feasible PoF with risk reduction techniques than to reject the SLA. In the case too many SLAs are rejected because the estimated PoF is too high, a low workload and thereby a low profit can result. Another important aspect to be considered is that providers

usually have a margin and no fixed percentage or value defining the profit they want to earn with the job execution.

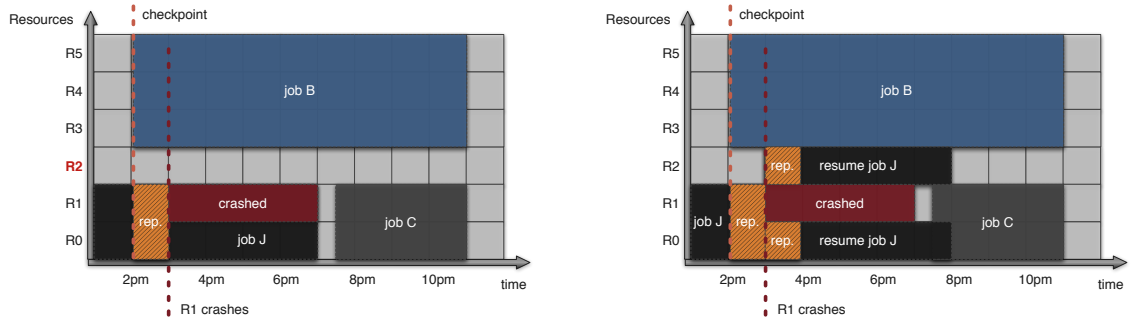
Planning FT-mechanisms to lower the feasible PoF already during the initial reservation process, enables providers to not strictly reject SLA bound jobs whose maximum accepted PoF cannot be provided. In addition to the FT-mechanisms planned in scope of risk reduction, policies may define to plan for all SLA bound jobs default FT-mechanisms. For instance, requiring checkpointing for all SLA bound jobs is a meaningful policy in order to avoid losses of computational steps. The checkpointing frequency might be fixed for all jobs or depend on the PoF or on the job length. Consequently, FT-mechanisms are either planned in order to lower the feasible PoF or to fulfil provider's policies. Note that the set of selectable FT-mechanisms depends on the capabilities of the underlying RMS (see Section 6.1.4).

Since any planned FT-mechanism lowers the PoF, also the planned default FT-mechanisms triggered by a policy are lowering the PoF and part of risk reduction. Hence, a differentiation of risk reduction mechanisms must not consider whether these are initiated because of a policy or a too high feasible PoF. Important in any case is that planned FT-mechanisms have to be considered in the PoF estimation. Either this is directly reflected in the risk assessment basic model or these reduce the initially determined basic risk, i. e. each enabled FT-mechanism reduces the basic estimated value which only considers failure rates of resources. Dependent on the capability of the RMS, following risk reduction means can be considered:

- initiate checkpointing
- extend job duration for restart
- make an FT-reservation
- hold a pool of spare resources
- plan a redundant job execution

Checkpointing is an important means for performing migrations since they enable to resume a job from an intermediate computation state instead of the initial one. Even if checkpointing is available and the data is stored somewhere in the network, snapshot data might not be available since either the first checkpoint has not been made before the resource outage or a storage or network defect forbids the usage of the data.[2] In those situations, the job has to be restarted from the beginning which equates a migration with a restart. This is in particular experiences a loss for parallel jobs, which are executed on several compute nodes, since a resource crash leads to a restart of the complete computation if no checkpoint is available. Consequently, the checkpointing frequency is important to be appropriately set. In some Grid environments contractors may explicitly ask for FT-mechanisms, such as checkpointing, in their SLA request. If checkpointing should be initiated in scope of risk reduction, checkpointing frequencies defined in the SLA request and in the policy have to be compared. Based on reliability estimations of high performance clusters [Schr 06, Raju 06] an optimal checkpointing frequency can be defined if no adequate information is available from the cluster itself. Observing up-times, down-times, and failure rates might then lead to adjusting the thresholds in the risk identification phase of the Grid Risk Management process.

[2] analysis about storage system failures or network outages can be considered to estimate failure probabilities. An example data source is [PDSI 08]



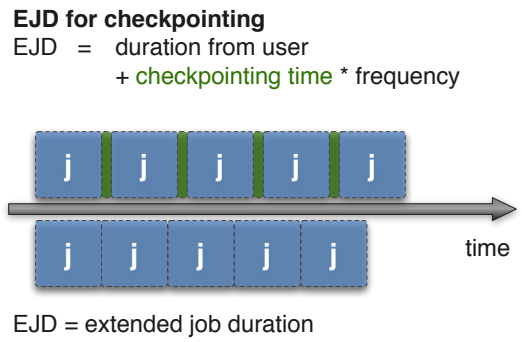
(a) Scenario of Node Crash of Parallel Job - R2 is alternative resource

(b) Repeating Steps of Job after Node Crash - Resuming on R2

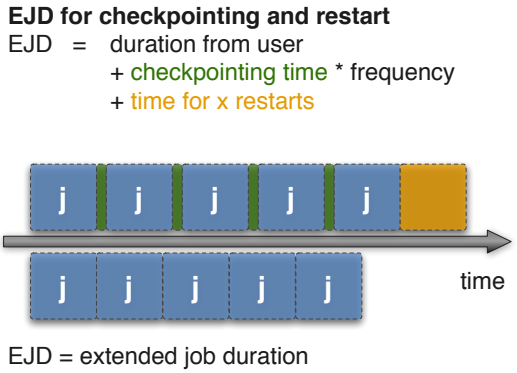
Figure 7.13: Additional Time is Needed to Handle Node Failure for Parallel Jobs

Since a planning based system requires an exact definition of the job duration, Section 6.1.4 pointed out that the scheduler extends the job duration for internal purposes if checkpointing is planned during the SLA negotiation. Thereby the requested execution time will be available and the resource reservation is long enough to make the snapshots according to the defined frequency. Even if the *Extended Job Duration* (EJD) enables checkpointing for parallel jobs, a migration which holds the deadline might be a challenge. The problem of performing a migration within the resource reservation results from the fixed job slot and is visualised in Figure 7.13. Consider a parallel job running on n nodes, i. e. it consists of n sub-jobs $s_i \in j$ where $1 \leq i \leq n$. If a node has crashed on which the sub-job s_k had been executed, the migration transfers the job data of s_k to another available compute node. The execution of the other sub-jobs $s_i \in j | i \neq k$ has to be reset to the latest checkpoint since all s_i are part of a parallel job implying an interaction with s_k. Consequently, the parallel job cannot be resumed before a suitable alternative resource has been found and allocated. Resources on which sub-jobs $s_i \in j | i \neq k$ had been executed, are reserved until t_e conforming to the sum of the planned start time and the EJD. The time required for the migration on the same cluster is in most cases negligible. However, the resource outage had required a repetition of the compute steps after the last checkpoint. In a planning based system the job reservation has a fixed length and accordingly, it has to be dynamically extended after a resource outage in order to provide the execution time as requested in the SLA (see Figure 7.13). Since in most cases this is not possible because of a high workload, during the SLA negotiation the job duration can be further extended in order to capsulate several restarts. This is similar to the extension for generating checkpoints. Historical observations about the number of resource outages can be used to determine the expected number of required restarts. The expected number should be planned. It is oblivious that the number depends on the job duration. Concluding, the EJD consists of the execution time requested by the user, time for generating checkpoints, and time for x restarts (see Figure 7.14b).

Even if the job duration is extended, the job can only be resumed if a suitable alternative resource is found. This might be difficult if only few suitable resources exist which conform to all constraints of the SLA. Furthermore, predictions about the availability of alternative resources are hard to make since these are based on the availability of the complete system and the SLA negotiations of other jobs. To reduce this uncertainty, reservations on additional



(a) EJD Resulting from Checkpointing Initialisation



(b) EJD Considering in addition Restarts

Figure 7.14: Checkpointing Initialisation Results in Extension of Job Duration

resources can be made, denoted as *FT-reservations*, and are assigned to one specific job. FT-reservations lowers the PoF for a job since they reserve dedicated spare resources. This strategy blocks resources for the usage of other jobs and consequently it should not be used as default. Rather it is a means to reduce the feasible PoF in order to achieve the upper bound of the contractor. If resources $r_1, \ldots r_n$ are reserved for the job execution and the estimated PoF is $x\%$ higher than the upper bound M, the scheduler can determine the number of required additional resources a in order to fulfill the upper bound m. The estimation how many additional resources are needed conforms to a loop, which adds one additional resource and re-estimates the PoF for the SLA. If resources have different PoFs, the scheduler has to preselect resources to be assigned to the FT-reservation. If all resources have the same PoF, the scheduler can determine first of all the number needed and then selects resources to reserve in scope of an FT-reservation. Assume, that for job j an FT-reservation covers resources $r_{d_1}, \ldots, r_{d_a}$. If any resource $r_i \in \{r_1, \ldots r_n\}$ fails during or before the job execution, the job uses a resource r_f of the FT-reservation, i. e. $r_f \in \{r_{d_1}, \ldots, r_{d_a}\}$. A resource outage of a resource $r_f \in \{r_{d_1}, \ldots, r_{d_a}\}$ does not harm the job execution as long as enough dedicated spare resources are available to compensate for resource outages of the set $\{r_1, \ldots r_n\}$. The loop which determines the number of required dedicated spare resources continues to add a resource until falling below the upper bound M. However, reserving many nodes in scope of an FT-reservation will not be profitable for providers. Consequently, it is necessary to control the maximum number of dedicated spare resources. Since this maximum value depends on the profit to be made and might be also influenced by customer policies, the negotiation module should determine the upper bound of additional resources. This determination depends on the profit margin the provider accepts as illustrated by the following example.

Example 7.4.1
Let a provider determines its price by the following formula, whereas $p = 0.3$ is the profit it wants to earn

$$\text{revenue} = [\text{total resource usage in h} \cdot \text{usage cost per h}] \cdot p \tag{7.18}$$

where *total resource usage in h* $=\#$ *nodes* $\cdot$ *duration in h* and the *usage cost* conform to external costs per CPU or node hour determined through a market price. Consider that the

provider accepts a profit between $20 - 30\%$ of claimed costs. Hence the profit margin is 10% and the money spent for additional resources a which are allowed to be reserved conform to $0.1 \cdot \text{price}$. Following,

$$a = \frac{0.1 \cdot \text{revenue}}{\text{internal usage cost per h}}. \tag{7.19}$$

It is important to use the internal cost for using a resource which may differ from the costs offered to consumers.

The negotiation module determines the maximum number of dedicated resources a based on the profit and, if necessary, on specific consumer's policies. This upper bound is forwarded to the scheduler which uses it as a stop condition in the loop when iteratively determining the number of required dedicated spare resources to fulfill the maximum PoF M. To work more efficiently, the scheduler can first of all check whether M can be achieved if a resources are additionally reserved. Then the loop would check a decreasing number of resources $a-1, a-2, \ldots$ and stops in step i if the estimated PoF is higher than M. An FT-reservation of the number of resources used in step $i-1$ is then made which conforms to the minimum number of additional resources resulting in a PoF $\leq M$. Note that reserving dedicated spare resources does not depend on the availability of checkpointing and migration techniques. If these are not supported by the RMS, dedicated spare resources can also be used for starting the job from the beginning. However, the lack of the basic FT-techniques requires to reserve longer time slots for the job execution in order to be able to complete a job even after a restart.

PoF Adaptation 7.4.1

The adaptation of the PoF is performed as shown in Listing 7.3.

Listing 7.3: Estimating PoF if x Dedicated Spare Resources are Reserved

```
1  double avgSuccess = sqrt[n](1 − feasiblePoF);
2  double avgPoF= 1 − avgSuccess;
3
4  int k = number of maximum allowed additional nodes;
5
6  . . .
7
8  for(i = 0; i ≤ k; i++)
9  {
10   double internalSum = 0;
11
12   for(j = i; j ≤ k; j++)
13   {
14     internalSum+ = k!/(j!·(k−j)!) · avgSuccess^j · avgPoF^(k−j);
15
16   }
17
18   probSuccess+ = n!/(i!·(n−i)!) · avgSuccess^(n−i) · avgPoF^i · internalSum;
19 }
20
21 If ((probSuccess > maxPoF) ∧ (k > 1)) then
22   test if k′ = k − 1 also results in a PoF lower than maxPoF.
```

Dedicated spare resources are beneficial for the job j to which they are assigned since these are always free to use for j. However, a resource outage affecting another job j' might cause an SLA violation even though free resources are available if all free resources are bound to FT-reservations. *Not* using resources belonging to an FT-reservation can be justified since it would affect the PoF of job j and the FT-reservation was made in order to accept the SLA of j. As a consequence, the RMS should also consider a pool of spare resources not previously assigned to any job. The pool has a fixed size which is configured by the system administrator and could be modified in scope of the risk identification process if observations point out that the number is not moderate. If a resource outage occurs, any job can use as many compute nodes of the pool as needed. Here, constraints can be made that only SLA bound jobs are allowed to use resources of the pool or that best-effort jobs may use the nodes as long as these are not needed by any SLA bound job. If several SLA bound jobs need resources from the pool, the strategy *First Come First Serve* (FCFS) can be applied. To resolve conflicts if more resources are needed than available in the pool, specific Risk Management activities can be performed. However, these applies for SLA bound jobs agreed and not for those which are under negotiation. Chapter 8 presents details. Taking a pool of spare resources into account for jobs under negotiation is only important in the context of the risk assessment. The initial PoF is reduced by considering the availability of spare resources useable in the case of a resource outage. In contrast to the FT-reservation, the availability of a spare resource is not guaranteed since too many resource failures could have occurred previously and the pool is empty. Hence, this concurrency of all SLA bound jobs has to be reflected in the risk assessment. Note that an SLA bound job j having an FT-reservation may also use resources of the spare pool. However, this should be only valid if the associated FT-reservation cannot compensate for the resource outages affecting job j. Concluding, holding a pool of spare resources is a general means to reduce risks, however, it is no job specific Risk Management.

PoF Adaptation 7.4.2

The difference between using dedicated spare resources and resources of the spare pool is that their availabilities do not only depend on their PoF, i. e. the probability that they have failed also. Any SLA bound job running in the system may use nodes of the pool in order to compensate for resource failures. Hence, this concurrency has to be reflected when estimating the probability that it can be compensated for x resource failures by using x spare nodes. To simplify the estimation of this probability, a workload of 100% of the system expect the pool is considered. Furthermore resources are assumed to be similar stable. Consider that the pool of spare nodes has a size of s. When estimating the PoF of an SLA violation, the probabilities to compensate for $1, \ldots, n$ resources has to be multiplied where n equals the number of resources to be used by the job.

$$\Pr(x \text{ nodes from pool can be used}) = \Pr(< s - x \text{ resources have failed}). \tag{7.20}$$

The estimation of the probability that less than $s - x$ resources of the cluster have failed considers the PoF of each resource as well as the number of resources of the cluster. If repair times are considered, the probability can be detailed by estimating that less than $s - x$ resources have failed during the last y hours, where y is the mean time to repair.

In the case the PoF of a job is significantly too high, a redundant job execution can be

planned instead of reserving many dedicated spare resources. If working with FT-reservations, a redundant job execution can be started if the number of additionally reserved resources is equal to or higher than the number of resources requested by the job. This case is however very unlikely. One reason to prefer a redundant job execution instead of reserving less dedicated spare resources is to execute the job on different clusters or even on different Grid sites. Thereby threats concerning the unavailability of the whole cluster, such as a power failure, breakdown of a key network component, or a natural disaster, will not definitely cause an SLA violation since the job execution on the other cluster might be successful. Since a RMS is responsible for the job management on one cluster, the gateway of the Grid fabric will delegate the jobs and their results appropriately (as introduced in Section 6.1). Another reason to plan a redundant job execution can be the unavailability of the FT-mechanisms checkpointing and migration. If a RMS does not support these, a redundant job execution will often be the only possibility to prevent SLA violations. Finally completing a job after starting it from the beginning after a resource outage will not be successful for tightly timed jobs.

PoF Adaptation 7.4.3
If job j is redundantly executed i times, p_i describes the PoF for an SLA violation for instance i. The p_is are independent and the total PoF for an SLA violation of job j is

$$\mathrm{PoF}(j) = \prod_i p_i \tag{7.21}$$

Planning mechanisms to reduce the risk is often more beneficial then rejecting the SLA offer because of a too high feasible PoF in ratio to the maximum PoF accepted by the contactor. Dependent on the capabilities of the underlying RMS, the initiation of checkpointing is valuable in order to save computing time in case of a resource outage. The job duration should be extended during SLA negotiations in order to gain time for checkpointing and several restarts without violating the runtime constraints defined in the SLA. In addition to checkpointing, reserving dedicated spare resources significantly reduces the PoF. These dedicated spare resources are always available for a job if these have not failed themselves. More profitable according to the system utilisation and wasted computation time is however the pool of spare resources which holds some resources free for the usage of any SLA bound job affected by a resource outage. This concept is more important in the Risk Management process handling jobs bounded by an SLA agreed, however, the availability of such a pool has to be considered in the PoF estimated during the negotiation. In case the RMS does not support checkpointing and migration, a redundant job execution is a valuable means to lower the estimated PoF. Planning any of these FT-reservation has to be performed during the SLA negotiation, lowers the feasible PoF, and can lead to the acceptance of the SLA.

7.5 Risk Acceptance, Avoidance, and Transference

The previous section presented techniques applicable in the context of risk reduction which is only one aspect of Risk Management. Note that risk reduction is also often denoted as risk mitigation. In addition to risk reduction/mitigation, Risk Management can apply one of the following strategies:

- risk acceptance,
- risk avoidance, and
- risk transference.

In the context of SLA negotiations, the provider accepts the risk if it commits to the SLA offer. The provider estimates the PoF for an SLA and calculates the risk as the product of the PoF and the penalty defined. Based on this information, the provider knows the business risk of accepting the SLA. Furthermore, it is able to define the revenue according to the risk and FT-mechanisms precautionary planned.

A rejection indicates that the risk avoidance strategy is followed. The provider is not willing to accept the maximum PoF or the penalty requested. Reasons can be either that the maximum PoF is not feasible when planning FT-mechanisms under a specific cost limit or that policies forbid the acceptance. By avoiding the risk, the provider will not loose money because of an SLA violation. However, it has to accept some risks, since otherwise its system will be idle or only used by best-effort jobs which is less profitable.

Figure 7.15: Risk Management Strategies in the Context of SLA Negotiation

Risk transference means that the risk is conveyed to another Grid provider. In the computational Grid risk transference is performed by outsourcing jobs. Following this strategy during the SLA negotiation phase is not sensible at this stage since the provider has the opportunity to reject the SLA offer. The only reason to outsource a job under negotiation are justified by customer policies. In the case the provider should accept the SLA of a specific customer (always or in that current situation) and it is not able to fulfil the maximum PoF, it can negotiate with another provider about the job execution.

The risk acceptance and avoidance correspond in the context of SLA negotiation to the acceptance or rejection of an SLA request. Risk transference is realised by outsourcing a job. A reason to outsource a not agreed SLA bound job might result from specific policies, otherwise following this strategy is not reasonable.

7.6 Reservation Process in the Renegotiation Phase

In scope of WS-Agreement Negotiation (WSAN), several offers and counteroffers can be exchanged between consumer and provider before agreeing an SLA. Consequently, reservations made might have to be adjusted if a counteroffer is received. Indeed currently only implementations of the WS-Agreement protocols are available [Wald 08, Batt 07], but new trends reflected also in the work of the Open Grid Forum's GRAAP group [OGF 08] highlight that a renegotiation is important after an initial SLA has been committed. This means that an SLA has been agreed by using WS-Agreement or WS-Agreement Negotiation and afterwards the renegotiation is performed based on the SLA previously committed by both parties. The previously committed SLA remains valid as long as the renegotiation has been successful since then the modified SLA comes into force. Such SLAs are also often denoted *dynamic SLAs* [Pich 08]. The WS-Agreement specification can be enhanced in order to define constraints of the renegotiation [Di M 07b], such as the number of modifications during the agreement's lifetime. Since specifications about the renegotiation protocol are not established yet, this section describes the handling of a new SLA request in scope of WS-Agreement Negotiation. The internal validity checks presented here are however the same for a renegotiation of dynamic SLAs. The only difference is that in a renegotiation the initial reservation is not cancelled before a modified SLA has been agreed. In the WS-Agreement Negotiation the previous SLA request must not be longer considered since no SLA has been agreed before.

The assumption is made that the provider has made an SLA offer to contractor C on the basis of an advance reservation r_C. Since a renegotiation step is considered, the contractor C has modified the SLA offer into a new SLA request and has sent it as a counteroffer to the provider or as a new request in scope of a renegotiation. The activities of the negotiation module in the renegotiation phase do not differ significantly from the initial phase. The only difference in the workflow concern the request sent to the scheduler. Dependent on the contractor's modification initial reservations could be adjusted according to the new SLA request instead of making a complete different resource reservation. Figure 7.16 depicts this situation. Section 7.6.1 – 7.6.4 presents an overview about possible adjustments in the SLA request and necessary actions from the RMS to generate a new SLA offer. Section 7.6.5 summarised the observations. Note that internally - after receiving the SLA request and before publishing a new SLA offer - the negotiation state of the provider will change to *advisory* under the terms of the WS-Agreement Negotiation (see Section 3.3.5).

7.6.1 Modified Revenue or Penalty Fee

If the contractor has modified the revenue or the penalty fee, the negotiation module has to compare the requested fees and the resultant cost for using the reserved resources. By taking into account customer and negotiation policies the negotiation module decides whether the

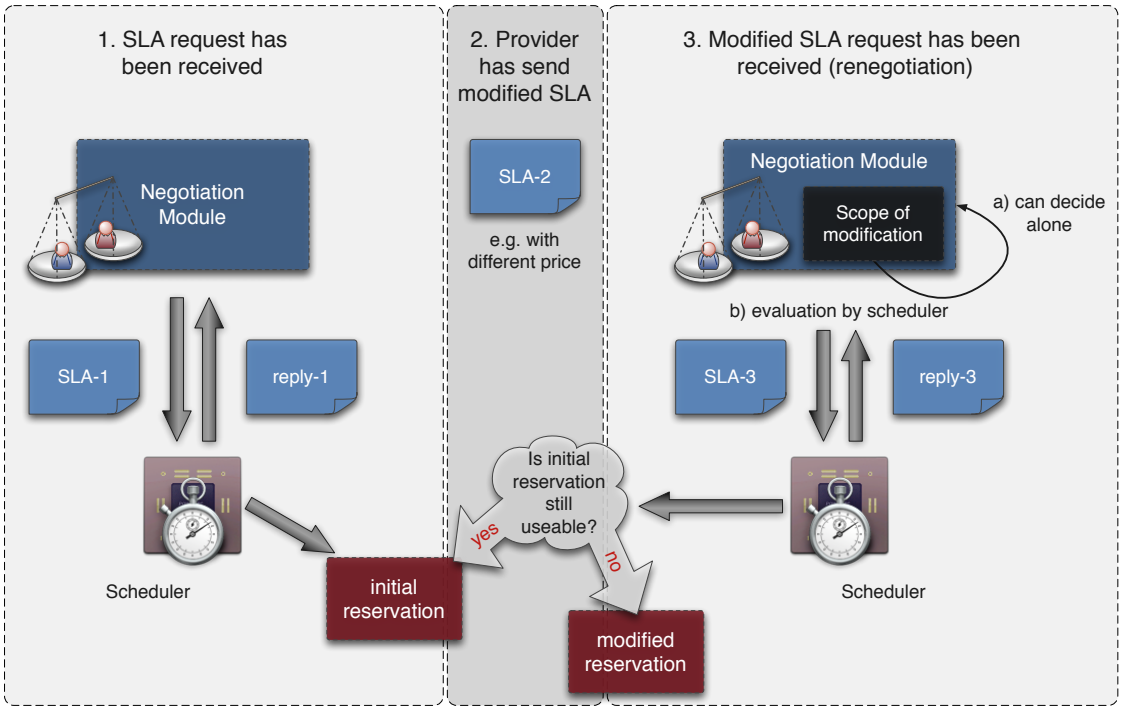


Figure 7.16: Negotiation and Renegotiation

profit is high enough to agree the SLA. Committing to the SLA implies the risk of loosing money because of the penalty fee and that reserved resources will not be available for other job executions. Resulting, the provider may generate an SLA offer with a lower revenue or a higher penalty fee than the recent one and which is probably higher or lower than in the SLA request. If the provider reacts in this way on a modified revenue and penalty fee request, the reservations of the initial process remain valid for the new SLA offer. Note that the provider will adjust the expiration time of the SLA offer since a renegotiation has appeared. An alternative way to handle a counteroffer with modified charge and penalty fee from the contractor is to make a new reservation and cancel the initial one. If the contractor asks for a higher penalty fee, the provider should only agree on this if it can also reduce the probability of an SLA violation. In order to reduce the PoF, either more stable resources should be used or the execution of FT-mechanisms should be planned precautionary, e. g. initiating checkpointing from the beginning of the job execution or making an FT-reservation (see Section 7.4). Note that reserving more stable resources is possible, if the provider operates different clusters whose resources have various stabilities or if significant differences in the behaviour of compute nodes of the same cluster have been identified. If different resource stabilities exist and these are not significant, the RMS should abstain from making a new reservation. Planning FT-mechanisms in scope of risk reduction instead of discarding the initial reservation reduces the effort of the RMS and is preferable. However in the case additional FT-mechanisms are too expensive in comparison to the revenue, the initial reservation r_C will be cancelled and the reservation process will be executed as in the initial phase. Only if the modified reservation (on more stable resources) is not too expensive, the provider generates a new SLA offer based on the new reservation.

If the contractor has asked for a lower revenue and the profit of the provider should not

decrease, the scheduler can make a new reservation on cheaper resources if these are available. Note that the cost for using a resource do not only depend on its stability, rather on the components and the vendor of resources. Accordingly, a cheaper reservation will usually imply the job execution on a different cluster, but does not necessarily imply that the probability of failure increases.

7.6.2 Modified Probability of Failure

In the case the customer asks in a counteroffer for a lower PoF, the provider reacts in the same manner as when a higher penalty fee is demanded: it tries to add FT-mechanisms in order to reduce the PoF for the job execution or it makes a new reservation on more stable resources. If FT-mechanisms are planned, the revenue usually increases since this additional service has to be paid. However, the negotiation module will decide according to policies whether the new SLA offer will really contain an increased revenue or whether the provider accepts a lower profit. In most cases the revenue will depend on the PoF, hence a lower PoF implies a higher revenue and a modification of the revenue is usually indispensable.

7.6.3 Modified Service Terms

If the contractor has modified service terms compared to the previous SLA request, the scheduler evaluates whether the initial reservation r_C still complies with the new requirements. It depends on which service terms has been modified whether the conformity of r_C to the SLA is checked by the scheduler or whether the check is omitted. Modified service terms defining resource characteristics might not conflict with an already made reservation r_C. If the initial SLA request does not specify all possible service terms or it has low requirements, modified resource characteristics might not transgress the validity of initial reservations. For example the contractor requested for a CPU speed of 800mHz in the initial SLA request and increases the speed constraints in a counteroffer, then the reservation r_C might conform to the new SLA request if resources are used having 2Ghz CPUs. Consequently dependent on the resource availability and resource requirements, reservations might be made on resources providing a better performance according to the CPU speed or the RAM size than requested by the contractor. In these cases a modified service term can be handled without making a new reservation.

If the contractor has extended the runtime of the job, in systems having a high workload the initial reservation r_C will often not be extensible; the scheduler makes a new reservation and omits to evaluate adjusting initial reservations. However, in the case of low system utilisation, the scheduler checks whether the initial reservation can be enlarged before cancelling them. If the runtime is decreased in the new SLA, the scheduler is able to adjust the initial reservation r_C in order to minimise the effort of the scheduler in the renegotiation step. Since a renegotiation addressing a decreased execution time will be rare, following this idea is better than making a new reservation. If renegotiations addressing a decreased runtime often appear, the scheduler has to be aware that such a modification might have bad influences on the schedule quality and the resource utilisation.

7.6.4 Modified Guarantee Terms/Service Level Objectives

If the contractor has modified guarantee terms, the negotiation module evaluates these in relation to the provider's policies and service functionalities. If guarantee terms/SLOs, such as the submission time, has been changed without affecting the planned execution slot, the scheduler does not have to make any adjustment of the reservations and the negotiation module is responsible for deciding whether the modification is valid. If however the planned execution slot in comparison to the earliest start time and deadline is considered in the PoF estimation, the modified PoF has to be estimated.

If guarantee terms concerning the support of FT-mechanisms have been modified, the scheduler plans appropriate FT-mechanisms precautionary or deactivates already planned ones. Note that when a contractor asks for a higher guarantee, the revenue usually increases. Furthermore, a guarantee term can be added for an already existing service term. Since not fulfilling service terms which are not bounded by a SLO does not lead to an SLA violation, such guarantee terms can be handled as guarantee terms which related SDT has been also inserted.

7.6.5 Conclusion

Counteroffers applied in scope of WS-Agreement Negotiation may lead to cancel the initial reservation and require to make a completely different one. In some cases the initial reservation can be modified, for example, if less resources should be used, lower or previously not defined resource requirements are demanded, or the execution time is shorten. If a lower PoF is required, precautionary planning additional FT-mechanisms may lead to SLA acceptance without discarding the initial reservation. The RMS can however not prevent to discard a reservation if significant modifications of the service or guarantee terms have been made. In this case the initial reservation is cancelled and a new one is made on the same or on a different cluster. Overall, such situations should be rare since a consumer usually knows the key requirements of their job when they send the initial request. Consequently, the most common case of a renegotiation concerns PoF, revenue, or penalty fee. The negotiation module can sometimes accept a modified revenue or penalty fee by setting these in relation to policies and profit margins. This implies that no interaction with the scheduler and no modification of the initial reservation is necessary.

Renegotiations performed after an initial SLA has been agreed, are handled similar to a counteroffer. The only difference is that the initial reservation must be valid until both parties have committed to a new SLA and consequently an initial reservation r_C must not be cancelled before.

7.7 Recapitulation of Risk Management During SLA Negotiation

The first field of application to support SLA provisioning by Risk Management is the SLA negotiation since at this stage the provider has to decide whether to accept or reject the SLA. The provider requires an estimation of the PoF for an SLA in order to know the involved

business risk when accepting the SLA. This information is essential in order to close the gap between SLA as a concept and as an accepted tool (as described in Chapter 2). The SLA contractor (end-user, broker, or another provider) will define in the SLA request an upper bound of the PoF they are willing to accept. The provider compares this maximum value with the feasible PoF it can achieve. The feasible PoF is calculated based on the reservation which is made by the scheduler during the SLA negotiation. Hence, the reservation process has to be modified in order to take into account PoFs. In this chapter two different approaches for a risk aware reservation making were presented. First in Section 7.2, a PoF-specific reservation process was described which balances PoF estimations and scheduling qualities. The main aspect is that in the first step the PoF is considered and afterwards the schedule quality. If various reservation sets have the same schedule quality, the decision depends then again on the PoF. Often providers prefer to keep established scheduling strategies and policies and a new complete reservation process would not be accepted. Consequently, Section 7.3 described the enhancement of arbitrary scheduling policies with risk awareness. This enhancement is performed on two different layers in order to separate Risk Management from the general scheduling and resource selection processes.

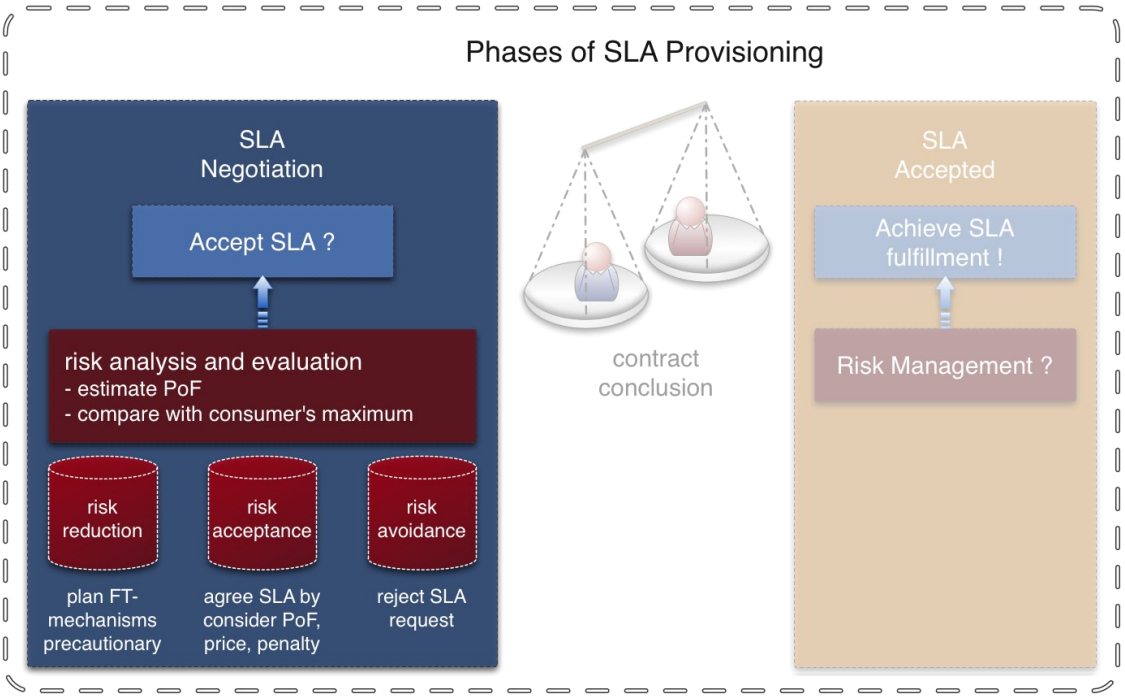


Figure 7.17: Addressed Risk Management During SLA Negotiation

In both reservation approaches the estimated PoF is compared with the maximum accepted one defined from the contractor. If the upper bound cannot be fulfilled, the provider may plan FT-mechanisms in scope of risk reduction (see Section 7.4). Dependent on the capabilities of the underlying RMS eligible means to reduce the risk are initiating checkpointing, extending the job duration to compensate for restarts of parallel jobs, making an FT-reservation – reserving dedicated spare resources –, holding a pool of spare resources, or planning a redundant job execution. It is important that during the risk reduction phase, the profit margin

is not left out of consideration since the initiation of any FT-mechanism results in additional cost lowering the provider's profit. In addition to risk reduction/mitigation, the provider can follow the strategy of risk acceptance, risk avoidance, or risk transference (see Section 7.5). Risk acceptance or avoidance corresponds to the final decision of SLA acceptance or rejection. Risk transference is achieved by outsourcing an SLA bound job, however, this does not makes sense if the SLA has not been agreed yet. Consequently, this strategy will not be considered in the following in the context of SLA negotiations. Figure 7.17 summarises the results of the Risk Management during SLA negotiation.

Chapter 8

Risk Management in Post-Negotiation Phase

Performing Risk Management for SLAs under negotiation differs from Risk Management in the post-negotiation phase in several aspects which results from the significant difference that during the SLA negotiation the PoF is primarily used as a decisive factor to agree or reject the SLA. In contrast to this, after the SLA has been accepted by the provider, the opportunity of risk avoidance, i. e. rejecting an SLA request, does not longer exist. This implies that the provider will definitely lose money if it does not fulfil its obligations as defined in the SLA. A further difference is that during the SLA negotiation planning and initiating *Fault-Tolerance* (FT)-mechanisms in scope of risk reduction can be performed without affecting other jobs. If other jobs would be affected, the SLA offer can still be rejected. Whereas after committing to the agreement, an initiation of FT-mechanisms usually has an impact for other jobs. Such an impact can rarely be avoided in systems with high workload.

Summarising the situation during an SLA negotiation compared with the one in the post-negotiation phase: risk serves primarily as a decisive factor for one job, FT-mechanisms usually do not effect other jobs, and risk avoidance is an opportunity. Risk Management activities in the post-negotiation phase should support the provider to work as profitable as possible even if less opportunities exist than during the SLA negotiation. Note that during the SLA negotiation applying risk avoidance must not be selected for each SLA request since then the provider would earn no money. Consequently some risks, i. e. SLAs, have to be accepted in order to have the chance to make profit.

The evaluation of which action should be initiated in the context of Risk Management for planned and running jobs has to take into account all supported FT-mechanisms as well as the option to *not* initiate an FT-mechanism. Initiating FT-mechanisms such as rescheduling, redundant job execution, or migration includes to considerately select an alternative resource. Hence, the purpose of integrating Risk Management in the post-negotiation phase focuses on the initiation of FT-mechanisms as described in Section 8.1. Estimating the impact of initiating an action is measured with the expected revenue which probably is made if initiating the FT-mechanism. The evaluation of initiating an FT-mechanism is performed in scope of a general Grid Risk Management process since a monitored event is the starting event. Differentiating between a monitored unstable resource state and a resource outage is sensible since the job can be successfully completed even on an unstable resource and consequently, an immediately handling is sometimes not necessary. Section 8.2 presents the principle of the impact estimation after a monitored unstable resource state which takes also into account that different resources might have different PoFs. This assumption enables to define general applicable impact estimation processes, however, a simplification can be made if all resources

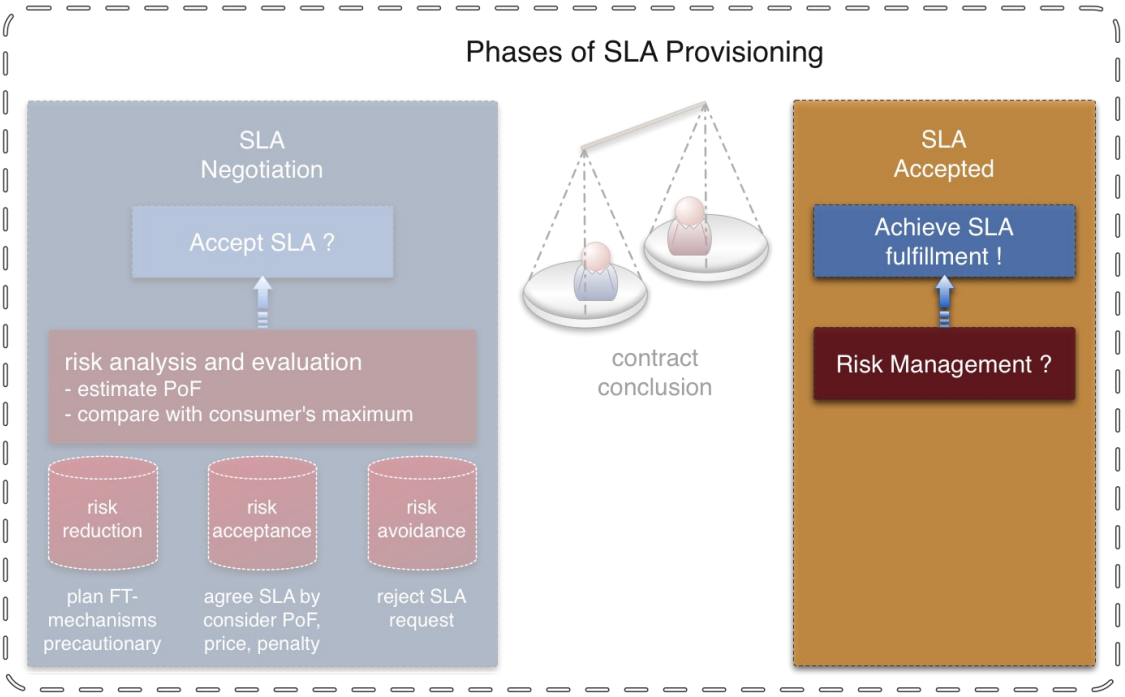


Figure 8.1: Risk Management in Post-Negotiation Phase ?

are similarly stable and have similar PoFs. Furthermore recursive checks of initiating FT-mechanisms is crucial in a complex system with a high workload and various SLA bound jobs. Section 8.3 describes both of these modifications of the evaluation process. The Risk Management performed after a resource outage has appeared, which affects an SLA bound job, is presented in Section 8.4. Section 8.5 gives a summary of the developments presented in this chapter.

8.1 Purpose of Integrating Risk Management Post-Negotiation

In order to maximise the provider's profit, the threat of paying penalties can be reduced by supporting FT-mechanisms to prevent SLA violations. In the context of resource failures, the most important and profitable FT-mechanisms are checkpointing, reserving dedicated spare resources for migration (FT-reservation), perform migration, or negotiate with other providers about outsourcing (see Section 6.1.4 and Section 7.4). Note that executing the job redundantly on an additional resource is also counted among FT-mechanisms, however, obviously, the multitude of requested resources reduces the profit of the provider. In addition to these job-specific FT-mechanisms, a pool of spare resource can be hold which make alternatives available for performing migration.

Migrations are an essential and cost-effective mechanism in SLA provisioning since additional own resources are only used when these are really needed. Furthermore, the job restarts not from the beginning, and consequently the already used computation capacities are not

wasted. The job execution loses the computation steps between the latest checkpoint and the resource outage, and consequently the checkpointing interval is very important. Since making a checkpoint immediately before the migration loses the least time as possible, *acting before* instead of *reacting after* a resource outage saves valuable time and will result in a higher success rate of preventing SLA violations. Consequently, identifying an unstable resource before its outage enables to make a checkpoint before the resource crashes. In particular, for jobs with run-times of days or weeks, the checkpointing frequency is often defined in an hour-interval. In this context, making checkpoint immediately before the job migration saves many computational steps which would have to be repeated otherwise. Acting before resource outages happen, increases the probability to fulfill SLAs even in the case of resource failures and reduces the amount of penalty fees which the provider has to pay in expectation.

If resources on the same cluster significantly vary in their stability, the scheduling should be aware of this when generating its mapping of jobs and resources. In this scenario selecting an alternative resource for resuming/restarting the job execution has to take into account the resource's PoF. That is why the decision of the RMS, which eligible FT-mechanisms should be initiated, has to consider resource stabilities and their schedules. The resource's schedule is a crucial aspect since a restart/resumption results in an impact on other jobs. Summarising, the evaluation process has to consider the resource stabilities, the schedules, and the effects when determining whether and which FT-mechanism should be initiated.

In the following Section 8.1.1 presents details about the monitoring of resources which is essential in order to identify unstable resource states. The Risk Management activities applicable for jobs in the post-negotiation phase are described in Section 8.1.2. Important is that the Risk Management in the post-negotiation phase can benefit from the risk reduction performed during the SLA negotiation. Section 8.1.3 considers the opportunities created by the previous risk reduction.

8.1.1 Monitoring

Monitored data can detect or at least give hints for the instability of a resource. Contemporary solutions such as Nagios [Bart 05] or a superior acting Unified Monitoring Framework [Lerc 06] support thresholds to automatically classify a monitored value as normal, critical, or warning. These support to send notifications when a state has become active or being valid for a defined time-interval. The notification can be used to initiate a risk re-assessment as part of the general Grid Risk Management process. Which thresholds should initiate a risk reassessment are system specific and have to be defined previously in the configuration phase from technical experts. In scope of the risk review process, an adjustment of these thresholds might be necessary if a warning or critical state appeared too late or too early in comparison with the monitored resource availability. In the monitoring system thresholds are defined in particular for dynamically changing resource characteristics. These significantly influence the stability of a resource and thereby the PoF of an SLA violation, for example it should monitor:

- CPU extent of utilisation
- RAM extent of utilisation
- free hard-disk space

- network extent of utilisation
- temperature of CPU and of RAM
- run speed of the CPU fan

The probability of a resource failure should be based on such monitoring data in order to take into account the resource stability. Since these values are dynamically changing, it uses either statistical values or forecasts. The PoF for an SLA also depends on data independent from a specific hardware and software: the availability of experts influence the *Mean Time To Repair* (MTTR), the PoF is influenced by FT-mechanisms like the pool of spare resources etc.

If a node outage is monitored, information about the resource failure has to be gathered in order to automatically determine which component has caused the failure. If identifying the cause cannot be performed automated, the monitoring information provides all information required to enable the administrator to identify the origins during a risk review process. The collected information might be useful to adjust the thresholds used in the state classification or to change the notification parameters if a state change of a monitored value has more or less consequences as previously expected.

8.1.2 Initiation of Risk Management

Risk Management can be initiated from two events. In the first scenario the scheduler is notified from the risk assessment about a modified PoF for a job in scope of the general Grid Risk Management process which is the result from an unstable resource state. In case of a resource outage, the scheduler detects this either by itself or by a notification of the monitoring system. This scenario is part of a targeted Grid Risk Management process and the scheduler initiates a risk assessment. One reason for a targeted Risk Management process is that after a resource outage an affected SLA bound job has never a negligible risk; consequently, a filtering process is not necessary. The risk assessment module estimates the modified PoF and notifies the requesting module which is in this case always the scheduler. The risk evaluation is performed by the scheduler itself instead of the risk assessment. This is sensible since the scheduler performs the resource mapping and in most cases only one job is affected by a resource outage for the same time. Consequently, the scheduler handles to prevent the SLA violation of the affected job by considering various strategies to initiate an FT-mechanism.

For each job planned or currently executed on a resource, which has been identified to be unstable, the scheduler has to decide whether and how it reacts. After a resource outage the affected running jobs have to be managed in order to prevent their SLA violation. Obliviously, jobs planned to start after the MTTR of this resource could remain scheduled on that resource since it probably is available after reparation or reboot. Which jobs can be kept out of consideration significantly depends on the forecasted time for the resource re-availability. Reassessing the PoF of SLAs will provide the scheduler a decision base for later planned jobs which must not be considered. Based on scheduler and customer policy, other jobs in the system, and available alternative resources useable for an FT-action, the provider can apply one of the following Risk Management strategies:

Risk Acceptance: *Initiate none fault-tolerance action for an affected job.*
The scheduler can decide to 'doing nothing' when the job is already executed redundantly

on another resource (since this FT-mechanism was negotiated or initiated earlier). If the job is not executed anywhere else, 'doing nothing' often indicates accepting to pay the negotiated penalty for the job and to take the loss of reputation into account. In spite of the fact that the penalty has to be paid for the affected job, this could be the most profitable solution, if a violation of other jobs would probably lead to a higher damage.

Risk Acceptance: *Wait for the resource reparation.*
If the job execution has already started and a checkpoint exists, the job will resume the execution from the checkpoint when the resource is available again. The job has to restart from the beginning if no checkpoint is useable. If the job was not running yet, the job execution will only be time-shifted on the same resource. Note that this strategy does only not result in paying the penalty fee if the resource is re-available so early that sufficient time remains to complete the job before its deadline.

Risk Reduction: *Initiate checkpointing, migrate, restart, or reschedule the job to an alternative resource.*
Which action is initiated depends on the status of the job execution and whether a checkpoint is available. The cost for using the alternative resource has to be taken into account by the scheduler. Furthermore the required time for data transfer has to be considered. In addition to initiating new FT-mechanisms, the risk reduction plans made during the SLA negotiation can be used (see Section 8.1.3)

Risk Transference: *Outsource the job to another provider.*
For this action a negotiation has to be initiated by developing an adequate SLA request. Outsourcing leads to additional cost for the resource provider but it prevent to accept the penalty fee and reputation loss. This action is only possible when the provider supports outsourcing and sufficient time for the data transfer remains.

8.1.3 Benefiting from Risk Reduction Performed during Negotiation

If the estimated PoF for a reservation has been to high in comparison to the maximum PoF requested by the user, the scheduler could plan precautionary FT-mechanisms if the resulting revenue would not be too low. In this section the following question should be answered: *Are the FT-mechanisms planned during the SLA negotiation beneficial for the Risk Management in the post-negotiation phase?*

Section 8.1.3 presented the following risk reduction techniques:

- initiate checkpointing
- extend job duration for restart
- make an FT-reservation
- hold a pool of spare resources
- plan a redundant job execution

If checkpointing has been initiated during the SLA negotiation, the checkpointing frequency can be increased in order to manage an unstable resource state. As discussed previously, the initiation of checkpointing results in an extended job duration. In planning-based systems an extension during or immediately before the job execution is often impossible if the system has

a high workload. If a checkpointing frequency x is necessary to achieve a PoF p, a previous initiated checkpointing is beneficial since the extension of the job duration is lower than if previously no checkpointing has been considered. This means that the job duration has to be only extended by the difference between the overhead resulting from the new initiated frequency and the previously initiated one. Since thereby the extension is smaller, it is more likely to be able to perform the extension. Furthermore, the initiated checkpointing is highly beneficial in the case of a resource outage. Finally the job must not start from the beginning if a checkpoint has been generated before. In this case the rescheduling needs to allocate an alternative resource for a shorter time period.

The extension of the job duration for a restart is analogous to extension for checkpointing. If an unstable resource state has been monitored, more restarts could be made available by increasing the runtime precautionary. The success of such an extension is more likely if previously the job duration has been already extended. However, the job duration extension for restarts should be expertly determined a priori in order to avoid dynamic modifications of reservation slots because of stability changes. Consequently, applying this risk reduction after the SLA negotiation should be only carefully initiated since a job extension has always bad effects on the schedule. After a resource crash, the job duration should not be modified since this crash was already planned within the previous job extension.

The availability of an FT-reservation simplifies the finding of alternative free resources for executing a migration, restart, or redundant job execution. The Risk Management in the post-negotiation phase can use these FT-mechanisms without affecting other jobs in the system since the resources are dedicated to handle problems of the associated job.

In contrast to an FT-reservation, the resources assigned to the pool of spare resources can be used by any SLA bound job in the system. Their availability simplify the search for an alternative resource since such a spare resource can be used without affecting any other job. As addressed in PoF Adaption 7.4.2 (see Chapter 7), the PoF estimation has to consider the concurrency of different SLA bound jobs for the usage of alternative resources assigned to the spare pool. Without considering the concurrency in the PoF estimation during the SLA negotiation, the PoFs of jobs j' would increase if a resource of the spare pool is used by a job j. This would lead to re-estimating the PoF of all SLA bound jobs j' running simultaneously with j. An evaluation of initiating an FT-mechanism for such jobs would be very complex. Hence, considering the concurrency in the PoF estimation during the SLA negotiation is preferred.

In the case that during the SLA negotiation a redundant job execution has been planned, the RMS has another instance of the job running or planned. Hence, even a resource outage affecting a job j need not lead to an SLA violation of j if no FT-mechanism is initiated. Only if at least one other instance is running of j, not initiating an FT-mechanisms might not result in an SLA violation.

The FT-mechanisms planned precautionary during the SLA negotiation simplify the handling of unstable resource states or resource outages in the post-negotiation phase. The benefits range from a better chance to extend the job duration as much as needed, over a higher probability to find a free and useable alternative resources without affecting any other job, up to the possibility that even without initiating an FT-mechanism the SLA would not be violated since a redundant job execution has been planned.

Important to note is that precautionary planned FT-mechanisms, consuming additional time or additional resources, can be deactivated after a resource outage. The deactivation is an important means in order to schedule a job which could not be scheduled otherwise. Obviously, the deactivation increases the PoF, however, often it enables to execute a job whose SLA would be violated otherwise.

8.2 Risk Management Initiated by Monitored Instability

Hardware characteristics can indicate that a resource becomes unstable and that its probability for a resource outage has increased. By monitoring the hardware behaviour, failures can be presaged with a given percentage of certainty. This enables a risk aware RMS to react before the resource outage instead of afterwards. The big advantage is that an earlier acting saves valuable time which might be essential in order to met the deadline. In particular handling before a resource outage is beneficial for RMS which support checkpointing since immediately before initiating a migration, a checkpoint can be generated. A challenge however is to not violate any more important SLA resulting from the initiation of an FT-mechanisms for the job affected by the unstable resource state. For commercial providers it is crucial to find the most profitable solution. Since each job has a PoF, focusing only on penalties and rewards of all jobs is not sufficient. This section presents evaluation measurements to compare the initiation of an FT-mechanisms with resulting impact on other jobs.

In the following the assumption is made that the monitoring system in the Grid fabric has identified a critical or warning state of resource r which initiated a PoF re-assessment for a currently executing job j, i. e. resource r is in the schedule of job j – defined as $r \in s(j)$ (see Section 6.1.2). The risk assessment has classified the PoF modification as not negligible and the scheduler is notified that the new PoF is higher than the PoF initially estimated during the negotiation – $P_f(j)$. The scheduler starts an evaluation whether and which FT-mechanism should be initiated. To make a final decision, the impact of possible FT-plans have to be compared, which is addressed by this section. Since the migration technique has most complex effects, their impact estimation is defined separately in Section 8.2.1. Section 8.2.2 presents the effects of all FT-mechanisms except migration. The process selecting whether, and if, which Risk Management activity (initiate which FT-mechanism or none) is the best appropriate one is described in Section 8.2.3. An example completes the description in Section 8.2.4. The work presented in this section has been published in [Voss 07a].

8.2.1 Evaluation of Migration Effects

If the job j runs on several compute nodes in parallel, an alternative node would replace r instead of rescheduling the whole job j when performing a migration. Since in environments with high workload executing/restarting the job on a free resource will not be often possible, a migration has an impact for other jobs which is essential to consider in the decision making.

The migration to an alternative resource r', which conforms to the resource requirements, results in the resource utilisation for job j in a time slot $[t_s; t_e]$. For single jobs this time slot can start at the earliest when the migration data is available on the resource, has to finish at the latest until the deadline, and its duration equals the remaining job execution time. The

scheduler assesses the remaining execution time through the specified execution time in the SLA as well as the runtime before the last checkpoint. According to these constraints the scheduler can determine for each eligible resource in which time slot the migrated job should be executed. Parallel jobs j have significant stricter time constraints since the sub-job s_k has to be executed together with other sub-jobs $s_i|i \neq k$ belonging to the same superior job j. In this case the reservation slot has to start immediately and end until the planned end-time on resource r. If finding such a time slot is not possible, the reservation of all sub-jobs $s_i|i \neq k$ have to be time-shifted to a point of time when enough resources are available. The time for transferring the checkpointing data is negligible if the job is resumed on another compute node of the same cluster. If a migration is performed to a resource on a different cluster, the data transfer will need significantly more time since the interconnects are slower. According to the model defined in Section 6.1.1 a RMS only manages one cluster and consequently a consideration of this is not necessary. Furthermore because of the slower interconnections, for parallel jobs it is often not reasonable to execute different sub-jobs on different clusters.

In the following the assumption is made that an appropriate time slot on resource r' has been found. The migration and resource utilisation of r' results obviously in least damage if it would be free. Sometimes it will be possible to obtain a free time slot for the job execution of j by shifting other job reservations in time, if this does not result in an SLA violation. Details have been presented in [Voss 06]. If a free time slot can be generated by time-shifts. The assumption is made that the changed probability of an SLA violation for time-shifted jobs is negligible since the underlying risk assessment model does not consider the buffer between the planned end-time and the deadline. Another reason justifying this assumption is when resource stabilities significant differ and influence the PoF estimation so that the modified time has negligible effects on the PoF. If the PoF is primarily influenced by different resource stabilities, in this case the PoF modification should be low since the time-shifted jobs will use the same resource.

For simplicity first of all the assumption is made that the determined time slot $[t_s; t_e]$ is either totally free or fully engrossed by one reservation, i. e. in $[t_s; t_e]$ there exits no reservation using $[t_s + x; t_e]$ with $x > 0$ so that $[t_s; t_s + x - 1]$ is free or assigned to another reservation. As the scheduler supports negotiation, is planning based, and risk reduction strategies may be defined during the SLA negotiation, different reservation types can be in the time slot: tentative or planned job-reservation as well as tentative or planned FT-reservation. Tentative reservations belong to jobs which are currently under negotiation. Note that only the planned job-reservation can belong to a job without an SLA and is performed in best-effort approach. Risk reduction plans of the negotiation phase, may have reserved dedicated spare resources or hold a pool of spare resources from which some are useable. Both cases conform to the situation that a resource has been found which has a free time slot.

In the following different cases are presented which are characterised by the used time slot on the migration target resource r'. The cases are presented in which $[t_s; t_e]$ is free, is booked with an tentative planned or FT-reservation, or it is booked with an tentative or planned job-reservation.

Case A: *r' has a free time slot in $[t_s; t_e]$*
The risk for paying penalties because of job j depends only on an SLA violation of job j since

the migration does not affect another job.

$$\text{risk}(\text{penalties}_j) \quad = \quad \text{penalty}(j) \cdot \Pr(\text{SLA(j) violated} | r' \in s(j))$$

Since this risk will be used in each following case, $\text{ROPenalty}_{r' \in s(j)}$ is denoted as $\text{risk}(\text{penalties}_j)$ for using a resource r' instead of resource r. If the free time slot was generated by time-shifts, the resulting risk increase of the affected jobs can be added to the value of $\text{risk}(\text{penalties}_j)$. If the job is in addition still executed after the migration on the originally used resource r, $\text{ROPenalty}_{r' \in s(j)}$ has to be multiplied with the probability that resource r fails. Running the job redundantly on the originally assigned resource r can prevent an SLA violation if the job is completed there successfully. In this case the migration has not been necessary since the originally assigned resource r did not fail before the job completion.

Case B: *r' has an tentative job-reservation for job j_e*
The risk for paying penalties because of job j depends on SLA violations of job j and job j_e as well as the probability that the SLA has been agreed for the tentative job-reservation.

$$\begin{aligned} \text{risk}(\text{penalties}_j) \quad = \quad & \text{ROPenalty}_{r' \in s(j)} \\ & + \text{penalty}(j_e) \cdot \Pr(\text{SLA}(j_e) \text{ agreed}) \cdot \Pr(\text{SLA}(j_e) \text{violated}) \end{aligned} \tag{8.1}$$

Considering the probability of an SLA violation makes only sense when the SLA has been agreed from both contract partners. That is why the probability of an SLA violation assumes that the SLA has been agreed. Accordingly, it is possible to multiply the two probabilities and do not have to consider $\Pr(\text{SLA}(j_e) \text{ agreed} \cap$ no FT-mechanism prevents SLA violation for $j_e)$. The probability of an SLA violation can be assessed based on forecasts for the number of suitable free resources.

Case C: *r' has an tentative FT-reservation for job j_e*
The risk for paying penalties because of job j depends on an SLA violation of job j as well as the probability that the SLA of j_e has been agreed and the tentative FT-reservation is needed.

$$\begin{aligned} \text{risk}(\text{penalties}_j) \quad = \quad & \text{ROPenalty}_{r' \in s(j)} + \text{penalty}(j_e) \cdot \Pr(\text{SLA}(j_e) \text{ agreed}) \\ & \cdot \Pr(j_e \text{ needs FT-reservation}) \cdot \Pr(\text{SLA}(j_e) \text{violated}) \end{aligned} \tag{8.2}$$

The probability that the FT-reservation is needed has only to be considered in the case the SLA from j_e has been agreed. The probability of needing an FT-reservation equals the probability of the resource failure of r'' on which the tentative job-reservation from j_e is mapped. And this probability is known to the scheduler or to the responsible module performing the comparison and decision process.

Case D: *r' has a planned job-reservation for job j_e*
The risk for paying penalties because of job j depends on an SLA violation of job j as well as on the probability that the SLA j_e is violated since the job cannot be executed on another resource, i. e. no FT-mechanism can prevent an SLA violation.

$$\text{risk}(\text{penalties}_j) \quad = \quad \text{ROPenalty}_{r' \in s(j)} + \text{penalty}(j_e) \cdot \Pr(\text{SLA}(j_e) \text{violated}) \tag{8.3}$$

Case E: *r' has an planned FT-reservation for job j_e*
Additional to the risk for paying penalties for job j itself, the risk for paying the penalty for job j_e depends on the probability that the FT-reservation is needed and on the probability that no FT-mechanism will prevent an SLA violation of j_e.

$$\begin{aligned} \text{risk}(\text{penalties}_j) &= \text{ROPenalty}_{r' \in s(j)} + \text{penalty}(j_e) \\ &\quad \cdot \Pr(\text{FT-reservation is needed for } j_e) \cdot \Pr(\text{SLA}(j_e)\text{violated}) \end{aligned} \tag{8.4}$$

Case F: *no migration of job j*
If no FT-mechanism is initiated, the risk for paying penalties caused by job j depends only on the SLA violation of job j if resuming its execution on resource r. In most cases the risk will be relatively high in comparison to the offered PoF and penalty, since otherwise the decision process for searching suitable migration targets would not have been initiated.

$$\text{risk}(\text{penalties}_j) = \text{penalty}(j) \cdot \Pr(\text{SLA(j) violated} | r \in s(j)) \tag{8.5}$$

A probability often used in these risk formulas is the probability that no FT-mechanism prevents an SLA violation of j_e. This probability expresses that either rescheduling or migrating j_e could not be initiated. Reasons are that this action would result in a higher damage than the SLA violation of job j_e or the initiated FT-mechanism would not prevent the SLA violation.

At the beginning of this section the assumption was made that one time slot type engross $[t_s; t_e]$. However, different time slots can exist in $[t_s; t_e]$ which cannot be time shifted to obtain a free time slot. In this case risk(penalties$_j$) is calculated by combining the appropriate cases described above. Since risk= $\sum_i \Pr(\text{event}_i) \cdot E(\text{loss}_i)$, the risk of two cases can be simply added if terms containing in both cases are not counted twice, such as the ROPenalty$_{r' \in s(j)}$.

8.2.2 Evaluation of Other FT-Mechanisms

The RMS of the provider will not only support migration as an FT-mechanism. According to the underlying model (see Chapter 6.1) the FT-mechanisms checkpointing, redundant job execution, rescheduling, and outsourcing are considered.

Generally, in a risk aware Grid fabric checkpointing does not have to be executed from the beginning of the job execution. It can be initiated if the probability of the resource failure increases. However, the necessary runtime extension is often not possible in a system with high workload (see Section 6.1.4). If checkpointing has been initiated previously, in the scope of Risk Management in the post-negotiation phase the checkpointing frequency can be adjusted. The redundant job execution on the same cluster is a solution which is only cost-effective if no migration is supported. Whereas, a redundant job execution on different clusters is reasonable since the data transfer for a migration might be too time-intensive. Reschedule a job will only be done for jobs which have not started yet or if no checkpointing/migration is supported since otherwise the job is migrated. Outsourcing can be performed in the scope of redundant job execution, reschedule, or migration. It just means that resources are used which do not belong to the resource provider itself. The key aspect for outsourcing is that SLA negotiations with other providers or brokers are necessary. The SLA request describing the job execution for outsourcing differs from the SLA the original provider has agreed with the end-user. For

example the deadline for receiving the results from the other provider should be earlier than the deadline the provider has agreed with the end-user. In case of migration the execution time will also differ from the original execution time of the job, since the job already run on the own resources.

Case G: *rescheduling job j to resource r'*
Rescheduling can be done in a time slot which consists of free parts, tentative or planned FT-reservations, or tentative or planned job-executions. Accordingly, the same formulas as in Section 8.2.1 presented for the migration can be used.

Case H: *redundant job execution of job j*
Initiating a redundant job execution on another resource r' in the scope of an FT-mechanism for already running jobs is only reasonable if no migration is supported. For not started jobs rescheduling instead of a redundant job execution should be preferred if an unstable resource state has been monitored. In this case a redundant job execution would result in higher costs because of the additional resource utilisation. If for a redundant job execution only free time slots are used, the risk equals to $\text{ROPenalty}_{r' \in s(j)} \cdot \text{P}(r \text{ fails before completion of } j)$, otherwise the same cases as for migration have to be considered.

Case I: *checkpointing on resource r*
Initiating checkpointing for a job j running on resource r does not use additional compute resources. The initiation results in an extended job duration and the usage of additional storage to save the checkpoints. The option only generate checkpoints and do not migrate is also possible either if the completion time of the job is very close or the risk is under the defined threshold in order to evaluate the migration possibilities[1].

$$\text{risk}(\text{penalties}_j) \quad = \quad \text{penalty}(j) \cdot \Pr(\text{SLA}(j) \text{ violated} | r \in s(j) \wedge \text{ checkpoint freq. } f) \quad (8.6)$$

The probability of an SLA violation, if using r and having a checkpointing frequency f, consists of the probability of a resource failure of r, the probability that r is not recovered early enough to prevent an SLA violation, and the probability that no migration can be initiated which uses the last created checkpoint and prevents an SLA violation. It is important to note that an initiated migration could also result in an SLA violation if the time does not suffice to complete the residual job until the deadline or a further resource failure occurs.

Case J: *outsource job j*
If job j is executed by another provider p_2, p_2's offered probability of an SLA violation is taken over. However, the penalty which is paid from provider p_2, denoted as j_{p_2}, in case of failure may be lower than the penalty the original provider has to pay to the end-user.

$$\text{risk}(\text{penalties}_j) = (\text{penalty}(j) - \text{penalty}(j_{p_2})) \cdot \text{PoF}(\text{SLA}) \quad (8.7)$$

In this and the previous chapter different formulas for calculating the risk caused by a job j are presented. The case differentiation is caused by various reservation types in the schedule and the FT-mechanism which could be initiated. The formulas used for estimating the impact of a migration can be also used for the evaluation of rescheduling or a redundant job execution.

[1] which will be determined in scope of risk review processes

However, in most cases not the same resources or time slots can be used to perform a migration or reschedule/redundant job execution. Additionally the costs for the FT-mechanisms are a crucial aspect to consider since these are significant lower for a migration than if the job has to start from the beginning.

8.2.3 Decision Process

Determining whether and which FT-mechanism should be initiated and if necessary selecting an appropriate resource, depends primarily on the risk of paying penalties caused by job j. The risks for each possible action are determined according to the formulas presented in the previous Sections 8.2.1 and 8.2.2. It is obvious that eligible resources have to satisfy the resource requirements. For a migration the requirements might be stricter than defined in the SLA, since the snapshot of the job has to be resumed in a nearly equal environment, thinking of versions and positions of libraries [Batt 08b, Batt 08a]. Non commercial resource providers will select the action, which results in the lowest risk. Commercial resource providers have to consider the revenue for the resource utilisation compared with the expected penalties to be paid. Accordingly, the action with the lowest risk must not be also the most profitable Risk Management activity. Commercial providers have to balance the risk(penalties$_j$) and the costs(resource usages). In the cost calculation not only the price for the resource usage itself, also the costs for the data transfer, as well as the usage of the FT-mechanism are considered.

8.2.4 Example

In the case resources are differently stable and unstable states can be identified by the monitoring system, the risk assessment initiate an evaluation of applying an FT-mechanism for jobs executed on such resources. The initiation is performed in scope of a general Grid Risk Management process by a notification sent from the risk assessment module to the scheduler or manager of fault-tolerance in the RMS. The importance of considering penalty fees combined with probability of failures in the decision which alternative resource to use, is demonstrated by an example in this section. Three compute resources r_1, r_2, and r_3 and two jobs A and B are considered which both use only one compute node. The consideration of a single-job using only one compute node is no limitation since the estimated measures would be similar if several compute nodes are used. The example is described through data presented in Table 8.1. Due to an unstable resource state of r_1, the PoF to complete job A on resource r_1 has increased, i. e. is now 50%, and the scheduler has been notified by the risk assessment module. This example shows the evaluation of a migration since the formulas are the same for rescheduling or a redundant job execution. The initiation of checkpointing is not considered since a good estimation is hard to make system-independent.

In order to enable a comprehensible comparison of the two jobs A and B, the assumption is made that both jobs have defined the similar SDTs and SLOs in their SLA. When the evaluation is initiated, the remaining time of the reservation slot is the same for both jobs, i. e. both reservations end at time t_p which ensures that the reserved time slots could be used by both jobs. The significant difference of A and B is the penalty fee (see Table 8.1).

Job Name	Penalty	Executed on	FT-reservation
A	500 €	r_1	none
B	1500 €	r_2	on r_3

Resource ID	Current PoF
r_1	50%
r_2	5%
r_3	17%

Table 8.1: Key Data of Example

If job A will not be migrated and is further processed on resource r_1 the risk for paying penalties caused by job A is assessed as described in case F – equation(8.5):

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{penalty}(A) \cdot \Pr(\text{SLA violation when using } r_1) = \text{ROPenalty}_{r_1 \in s(j)} \\ \Leftrightarrow \text{risk}(\text{penalties}_A) &= 500\,€ \cdot 0.5 = \underline{250\,€} \end{aligned} \tag{8.8}$$

Pr(SLA violation when using r_1) equals the PoF of using r_1 and the penalty of A is defined as 500 €.

If job A is migrated to resource r_3, the planned FT-reservation for job B will be cancelled. This cancellation increases the risk of paying the penalty of job B which is added to the risk of paying penalties according to Case E – equation(8.4):

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{ROPenalty}_{r_3 \in s(A)} + \text{penalty}(B) \cdot \Pr(\text{FT-reservation is needed for } B) \\ &\quad \cdot \Pr(\text{no FT-mechanism prevents SLA violation for } B) \\ \Leftrightarrow \text{risk}(\text{penalties}_A) &= 500\,€ \cdot 0.17 + 1500\,€ \cdot 0.05 \cdot 0.5 = \underline{122.50\,€} \end{aligned} \tag{8.9}$$

$\text{ROPenalty}_{r_3 \in s(A)}$ is smaller than $\text{ROPenalty}_{r_1 \in s(A)}$ since r_3 has the lower PoF from 17%. The probabilities used in the second term have to be discussed in detail. First of all, more information is presented on the probability that no FT-mechanism prevents the SLA violation for B, afterwards the probability that the FT-reservation is needed for B.

If job A is migrated to resource r_3, the planned FT-reservation for job B is cancelled. If it is necessary before the job completion to migrate B to another resource, the only usable resource is r_1 in this example. The migration will not prevent an SLA violation of job B if resource r_1 fails. Consequently, the probability that no FT-mechanism prevents an SLA violation conforms the PoF of resource r_1, i. e. 50%. If the job B is migrated to resource r_1 and additionally resumed on resource r_2, no SLA violation might occur if r_1 fails but r_2 completes the job execution. However, only the probability, that no FT-mechanism prevents an SLA violation, has to be considered and not the probability, that the SLA is violated of job B. The migration of job A does only affect that the planned FT-reservation for job B is cancelled and as a consequence another resource has to be used when the initiation of an FT-mechanism for job B is required.

The probability that the planned FT-reservation is needed, conforms in a simplified approach to the PoF of resource r_2 on which job B is executed. This is valid since the FT-reservation is only needed if the resource r_2 fails. However, it has to be kept in mind that when an unstable resource state has been monitored, the migration to an FT-reservation can be initiated before a resource failure and not after it. In this case the FT-reservation would be used for resuming the job if the PoF for a resource is higher than a defined threshold T. Accordingly, the probability that the FT-reservation is needed, conforms to the probability that the PoF from resource r_2 transcends the threshold T during the remaining execution time of job B. The threshold T is defined in the configuration of the Risk Management by experts and in risk review phases it will be adjusted in an automated manner. The risk review phase evaluates on historical events whenever a migration has been necessary, i. e. the resource has failed before the job has completed.

In this example threshold T was set at 50%. The probability that the PoF of resource r_2 increases from 5% to 50% before job B has been completed is determined according to the historical behaviour of resource r_2 as well as the remaining execution time of job B. Let this probability be 10%, then the probability that job B has to use resource r_1 equals to the sum of the probability that r_2 fails, i. e. 5%, and that r_2's PoF has increased over T, i. e. 10%. In comparison with equation (8.9) this sum is used in the second term and the risk of paying penalties caused by job A when using r_3 is:

$$\text{risk}(\text{penalties}_A) = 500\,€ \cdot 0.17 + 1500\,€ \cdot (0.05 + 0.1) \cdot 0.5 = \underline{197.50\,€} \tag{8.10}$$

The last possibility for a migration target for job A is resource r_2 on which job B is executed. In the case job A is migrated to r_2, the job execution of job B has to be stopped there B is migrated to the planned FT-reservation on resource r_3. This possibility was described in case D and the risk for paying penalties caused by job A is:

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{ROPenalty}_{r_3 \in s(A)} + \text{penalty}(B) \\ &\quad \cdot \Pr(\text{no FT-mechanism prevents SLA violation of } B) \\ \Leftrightarrow \text{risk}(\text{penalties}_A) &= 500\,€ \cdot 0.05 + 1500\,€ \cdot 0.17 = \underline{280.00\,€} \end{aligned} \tag{8.11}$$

Due to the low PoF of r_2 being 5%, the risk for paying penalties because of an SLA violation of job A ($\text{ROPenalty}_{r_2 \in s(A)}$) is low when it is executed on resource r_2. The negative aspect of this solution is the significantly increased risk of paying the penalty of job B. The probability of an SLA violation when job B uses resource r_2 was 5%. If the execution of job B will be stopped on resource r_2 because job A is migrated there, the initiated FT-mechanism for job B will be the migration to its planned FT-reservation. The alternative resource r_3 has however a PoF of 17%.

Table 8.2 presents the evaluation results of this example. It shows that the best solution (based on historical information) is to resume the job execution of A on r_3 which implies to cancel the planned FT-reservation of job B. To cancel the planned job execution of B is the worst case since job B has a triply higher penalty fee than job A and the migration target of B has a PoF which is six times higher than the current used resource r_2.

Table 8.3 presents the evaluation results if the penalties for job A and B are switched, i. e. $\text{penalty}(A) = 1500$ and $\text{penalty}(B) = 500$. The PoFs and the current resource allocation

Resume job A on	risk(penalties$_A$)
r_1	250.00 €
r_2	280.00 €
r_3	197.50 €

Table 8.2: Evaluation Results

Resume job A on	risk(penalties$_A$)
r_1	750.00 €
r_2	292.50 €
r_3	160.00 €

Table 8.3: Evaluation Results with Interchanged Penalties

before the evaluation have not changed to the first example (see key data in second tabular in Table 8.1). In this example the less risky solution is to resume job A on resource r_2 and migrate job B to its planned FT-reservation on resource r_3.

The interchanged penalty fees show that a consideration of these are essential in the initiation of FT-mechanisms. However, these have to be combined with PoFs since jobs or resources may have significant differences in their success or stability.

8.3 Modification of Evaluation Formulas

Most formulas presented in Section 8.2.1 and Section 8.2.2 contain probability estimations regarding the resource stability. This can be often simplified in homogeneous clusters as described in Section 8.3.1. In contrast to the simplification of the formulas, the process of evaluating an FT-mechanism has to be detailed. In the previous section the success probability that an FT-mechanism prevents an SLA violation of affected jobs is considered. To be more precise, a recursive evaluation should be performed. Section 8.3.2 presents the usage of recursive checks which is exemplarily demonstrated in an example in Section 8.3.3.

8.3.1 Simplification of Evaluation

The scenario in which the evaluation process is initiated considers that a job j is running on a resource r being in an unstable state such that the risk assessment module has notified the scheduler. In order to determine the risk paying penalties caused by job j, all formulas are based on equation (8.1) defining the risk of paying penalties if resource r' is used instead of r:

$$\text{ROPenalty}_{r' \in s(j)} \quad = \quad \text{penalty}(j) \cdot \Pr(\text{SLA(j) violated} | r' \in s(j))$$

Compute nodes of a homogenous cluster consist of the same hardware and are in most cases similarly configured. Since their behaviour and resource stabilities are similar, resources usually have the same PoF. To simplify the formulas presented in Section 8.2.1 and Section 8.2.2 it can be acted on the assumption that both unstable resource states are rare and identified separately and the probability of an SLA violation for j depends not on different resource stabilities of r and r'. Consequently, the ROPenalty can be removed from the formulas which simplifies and speeds up the calculation.

8.3.2 Recursive Checks

The evaluation of initiating an FT-mechanism for a job j by using resource r', often has an impact on other jobs j_e running or planned on r'. Most formulas in Section 8.2.1 and Section 8.2.2 take this into account by the probability that no FT-mechanism prevents an SLA violation of job j_e. Either these values are statistically observed or possible FT-mechanisms applied on j_e are evaluated. In the formulas presented the probability that an SLA is violated is not further specified, the example in Section 8.2.4 considered the activities which would be performed for the affected job j_e. However, initiating a chain of FT-mechanisms is not reflected in the previous formulas. In the case that *no* other alternative resource would be free to execute job j_e, the probability of an SLA violation of j_e would be 100%. Since the RMS is however capable of performing FT-mechanisms, the questions arise: *How are chains of FT-mechanisms evaluated? Whether and which restrictions exist that FT-mechanisms are not initiated for jobs j_e affected by an initiation of an FT-mechanism for job j?*

These questions are answered in the following two sections. The first one is addressed by the concept of the recursive check process which is presented in the Section 8.3.2.1. Section 8.3.2.2 addressed the second question by defining the termination condition. The termination condition specifies the restrictions which exist for applying a chain of FT-mechanisms and prevents endless loops.

8.3.2.1 Concept of Recursive Checks

Recursive checks can be integrated by modifying the estimation of the risk of penalties caused by job j. In order to consider a chain of initiating FT-mechanisms for different affected jobs, the risk of penalties caused by those affected jobs have to be added to risk(penalties$_j$). The modification of equation (8.3) describing Case D (the alternative resource r' has a planned job reservation for job j_e in the considered time slot) generalises the estimation [Voss 08a]:

$$\underline{\textbf{Case D:}} \quad \text{risk}(\text{penalties}_j) = \text{ROPenalty}_{r' \in s(j)} + \text{penalty}(j_e) \cdot \Pr(\text{SLA}(j_e)\text{violated})$$

$$\overset{\text{generalised}}{\Longrightarrow} \quad \text{risk}(\text{penalties}_j) = \text{ROPenalty}_{r' \in s(j)} + \text{risk}(\text{penalties}_{j_e}) \tag{8.12}$$

All formulas presented previously (equations (8.1) - (8.4)) can be modified to be expressed by using the risk for paying penalties of the affected job j_e. A profitable solution requests for not generally cancelling affected jobs and initiate an FT-mechanism for affected jobs, if it is profitable in expectation. To estimate the probability that an FT-mechanism can prevent

the SLA violation of job j_e, statistically determined probabilities could be used. However, to define the probability precisely, an analysis of initiating FT-mechanisms for j_e results in a recursion. Important is that a chain of initiating FT-mechanism has to be profitable since otherwise paying a penalty fee results in a lower loss for the provider. As a consequence, the decision process and termination condition of the recursive evaluation are presented in the following.

Applying an FT-mechanism results in cost which decreases the provider's profit. Cost of a service technology such as checkpointing or migration is based on their development and maintenance. Since this working effort is relationally small in comparison to the number of service utilisation, cost for developing and maintaining the service technology is negligible. However, the execution of an FT-mechanism results in additional cost which is not negligible:

Making checkpoints leads to the consumption of storage as well as of network capacity since the data is stored not locally on the same compute node (otherwise the checkpoint would not be available in case of a node outage). Cost for a migration results from the data transfer of the checkpoint and depends on the impact on other jobs. If, as a consequence of a job migration, the probability of an SLA violation increases for some job(s) j_e, the difference between the risk for paying penalties with and without the migration is an additional component of the total cost for migrating. Totally the costs for using an FT-mechanism are (whereas C is used for costs):

$$\begin{aligned} C\,(\text{performing FT for } j) \;=\; & C\,(\text{utilising and applying FT-Technology}) \\ & +C\,(\text{impact for jobs } j_e) \end{aligned} \tag{8.13}$$

In contrast to the cost, the initiation of an FT-mechanism has also a measurable benefit. This can be calculated easily by the risk for paying penalties with $(j|FT)$ and without $(j|\neg FT)$ the initiation of the selected FT-mechanism caused by job j.

$$\text{benefit}(\text{FT}_j) = \text{risk}(\text{penalties}_{j|FT}) - \text{risk}(\text{penalties}_{j|\neg FT}) \tag{8.14}$$

The sign of the benefit calculated by equation (8.14) is of crucial importance. If the benefit for job j is not positive, no FT-mechanism will be initiated for jobs j and j_e and the risk of an SLA violation is accepted. This case will occur if either in expectation other jobs in the system are more profitable to execute or the resource instability is not significant enough in order to perform an FT-mechanism.

Calculating the benefit for a job j_e, which is affected by the FT-mechanism of job j, has to be considered in more detail if failure probabilities of resources differ. In this case the benefit for j_e is positive, if the job is mapped to a more stable resource. Mapping j_e instead of j to a more stable resource might be done either if resource and time constraints of j and j_e differ or j_e has a higher penalty. Then the initiation of the FT-mechanism for j combined with the FT-mechanism for job j_e results in a benefit for job j_e which would not be taken advantage of if the evaluation process for j has not been initiated.

Calculating $\text{benefit}(\text{FT}_{j_e})$ equals the risk for paying penalties caused by j_e, if the FT-mechanism for j_e is initiated, minus the risk for paying penalties caused by j_e, if *no* FT-mechanism is initiated. The important aspect is that $\text{risk}(\text{penalties}_{j_e|\neg FT})$ equals the penalties caused by j_e if no FT-mechanism is initiated for job j. Therefore this risk conforms to the risk before the recursive check was executed.

8.3.2.2 Termination Condition of Recursive Checks

The recursive checks can be either terminated if the penalty of job j is so low that is even less than the risk for paying penalties caused by any affected job j_e. Furthermore, the recursive evaluation can stop if one FT-mechanism uses a free time slot or if in expectation the total cost for the (chain of) FT-mechanism(s) is higher than the total benefit. Total cost and total benefit have to be considered since for each job j_e, involved in a chain of FT-mechanisms, specific cost occur and benefits or disadvantages can be noted. Using formulas the recursive checks terminate if one of the following cases is valid:

$$\text{risk}(\text{penalties}_{j|\neg FT}) \leq \text{risk}(\text{penalties}_{j|FT}) \tag{8.15}$$

$$C\,(\text{impact for other jobs}) = 0 \tag{8.16}$$

$$\sum_{j'} \Big(\text{benefit}(\text{FT}_{j'}) - C\,\big(\text{performing FT for } j'\big)\Big) \leq 0 \tag{8.17}$$

The termination conditions reflect the ratio of benefits and cost for performing an FT-mechanism. It is important to consider this ratio since the additional resource consumption (of storage or network capacity) reduces the profit. Even more important is the risk increase of job j_e if performing an FT-mechanism. The recursive checks can be stopped if a free time slot has been found since then the latest performed FT-mechanism does not affect any other job and the chain is valid. If the resulting cost would be too high in comparison with the penalty fee of job j, the chain is not proportionally and should not be initiated. The following section presents an example which applies recursive checks and the termination condition.

8.3.3 Example of Recursive Evaluation

Consider eight compute resources $r_1, r_2, \ldots, r_8$, jobs $A, B, \ldots, E$, and that an unstable resource state for resource r_1 has initiated the evaluation at time 11:30am when the schedule as presented in Table 8.4 was valid. The table shows for each job the PoF and penalty negotiated. Furthermore, a state differentiation is made between running and planned jobs. Note that all jobs are SLA bound and agreed, no tentative reservations under negotiation exist in this schedule. The deadline is important, since the SLA is violated if a job runs longer than the deadline. In the schedule of 11:30am all deadlines for the jobs are met identifiable by comparing the end times listed in the state and the deadline. In order to evaluate whether the initiation of an FT-mechanisms, such as rescheduling or migration, is possible, the remaining execution time is of crucial importance. The used resources are listed for each job in the table, for an easier understanding the schedule is depicted in Figure 8.2. Note that for job E dedicated spare resources have been reserved during the SLA negotiation: resources r_7 and r_8 are assigned to a planned FT-reservation for job E.

This example applies the simplification presented in Section 8.3.1 by making the assumption that all resources are similar stable and that each resource has the same probability to fail during the job execution. Thus, Pr(SLA(j) violated$|r' \in s(j))$ does not depend on r'. Furthermore, the assumption is made that checkpointing is performed on job A. The latest checkpoint

	A	*B*	*C*	*D*	*E*
PoF	10 %	5 %	3%	15 %	8 % (12%)
Penalty	500 €	700 €	3000 €	10 €	5000 €
State	running -1:30pm	running -2pm	running -1:30pm	planned 2pm -4pm	planned 1:30pm -10pm
Deadline	1:30pm	4pm	1:30pm	4pm	10pm
(remain.) Exe-time	1h30	2h30m	2h	2h	20h30m
Use spare nodes:	r_1, r_2, r_3	$r_4, r_5,$	r_7, r_8	r_8	r_2, r_3, r_4, r_5 (r_6, r_7)

Table 8.4: Schedule at 11:30 am

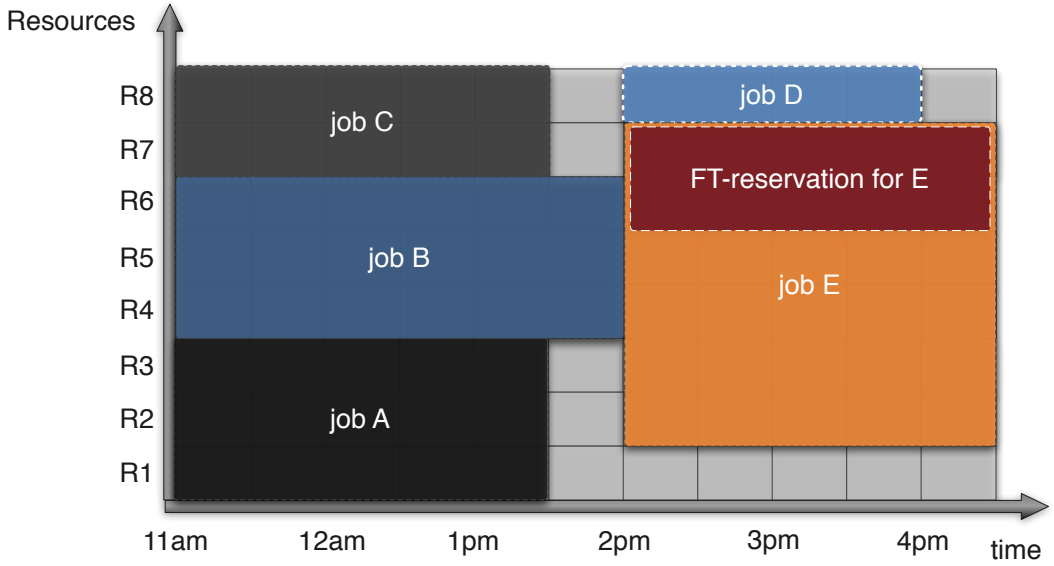


Figure 8.2: Example Schedule at 11:30 am

made was not longer than 30 minutes in the past. So job A can be completed until its deadline 1:30pm, if it is restarted at 11:30am with a maximum remaining execution time of 1 hour 30 minutes plus the repetition of lost computation steps of max. 30 minutes.

Since resource r_1 became unstable, the evaluation of FT-mechanisms is initiated for job A. Note that the cluster considered only consists of eight nodes and that with the schedule presented in Table 8.4 it is not possible to generate a free time slot by shifting the jobs. Finally all jobs have tight deadlines which would be violated through time-shifting.

In this section only the FT-mechanisms migration, reschedule, and restart are considered. Since the success of outsourcing depends on the systems, workloads, and negotiation policies of other providers, the success of an SLA negotiation with other providers is hard to estimate. To show the benefits of performing recursive checks, Section 8.3.3.1 presents the expected loss

of the provider if no recursive checks are performed. The process and results of a recursive evaluation are shown in Section 8.3.3.2.

8.3.3.1 No Recursive Evaluation

In the first step the evaluation of FT-mechanisms does not explicitly consider a recursive initiation. If this is not performed, jobs affected by a migration from job A will fail and the penalty has to be paid (with a probability of 100%). The following possibilities for initiating a migration for the sub-job of A used resource r_1 exists:

1. Initiate no FT-mechanism for job A, let the SLA be violated when using resource r be 90%.

$$\text{risk}(\text{penalties}_A) = \text{penalty}(A) \cdot 90\% = \underline{450\ €} \tag{8.18}$$

2. Using one resource of $\{r_4, r_5, r_6\}$ for A; B fails if no FT-mechanism is initiated, i. e. $\text{risk}(\text{penalties}_B) = \text{penalty}(B) \cdot 100\%$

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{penalty}(A) \cdot 10\% + \text{risk}(\text{penalties}_B) \\ &= 500\ € \cdot 10\% + 700\ € \cdot 100\% \Rightarrow \underline{750\ €} \end{aligned} \tag{8.19}$$

3. Using one resource of $\{r_7, r_8\}$ for A; C fails if no FT-mechanism is initiated

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{penalty}(A) \cdot 10\% + \text{risk}(\text{penalties}_C) \\ &= 500\ € \cdot 10\% + 3000\ € \cdot 100\% \Rightarrow \underline{3050\ €} \end{aligned} \tag{8.20}$$

If no recursive checks of initiating FT-mechanisms are performed, the lowest risk for paying penalties caused by job A is to do not migrate job A and resume the job execution on resource r which probably will result in violating A's SLA. In the case the evaluation would be performed after a resource failure of r, the SLA violation of A is accepted from the RMS.

8.3.3.2 Recursive Evaluation

A recursive analysis of initiating FT-mechanisms can lower the risk of paying penalties caused by job A for equation (8.19) and (8.20), because then the probability of an SLA violation for job B or C will not be 100%. First of all initiating an FT-mechanism for job B is considered. Since job B is already running, a migration should be done instead of restarting or rescheduling it. If the scheduler decides to migrate job B, in order to use resources of B for job A, it makes a checkpoint of job B immediately before stopping the job execution of B in order to do not loose any computation steps.

1. Initiate no FT-mechanism for job B

$$\text{risk}(\text{penalties}_B) = \text{penalty}(B) \cdot 100\% = \underline{700\ €} \tag{8.21}$$

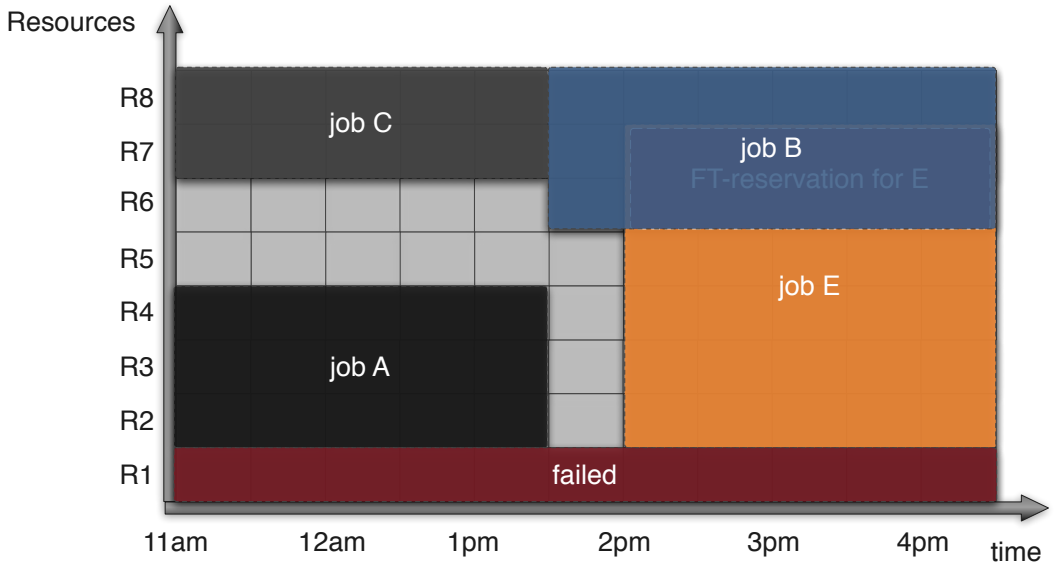


Figure 8.3: Example Schedule after Initiation of Migration for A and B

2. Time-shift job execution of B to 1:30pm, using resources $\{r_6, r_7, r_8\}$ for B; D fails since no FT-mechanisms will be initiated for it, and the risk for paying penalties caused by E will increase because the additionally reserved resources will not be free for E in the case of failure.

$$\begin{aligned} \text{risk}(\text{penalties}_B) &= \text{penalty}(B) \cdot 5\% + \text{risk}(\text{penalties}_D) + \text{risk}(\text{penalties}_E) \\ &= \text{penalty}(B) \cdot 5\% + \text{penalty}(D) \cdot 100\% \\ &\quad + \text{penalty}(E) \cdot \Pr(E \text{ needs FT-reservation}) \cdot \Pr(\text{SLA}(E)\text{violated}) \\ &= 700\ € \cdot 5\% + 10\ € \cdot 100\% + 5000\ € \cdot 4\% \cdot 100\% \Rightarrow \underline{245\ €} \end{aligned} \tag{8.22}$$

The first case conforms to the estimation used in the previous section. The value of the second case is significantly lower, since the not executed job D has a very low penalty and the usage of the FT-reservation of job E does not have a high impact on the risk for paying penalties caused by E. The estimation considers that without an FT-reservation the probability of an SLA violation has been 12 % for using four resources. Accordingly, the probability that one or two resources of the four reserved resources r_2, r_3, r_4, r_5 fails is estimated to be 4 %.

Combining the migration of job A executed on resource r_1 with the time-shift and migration of B has therefore the total risk for paying penalties caused by A :

$$\begin{aligned} \text{risk}(\text{penalties}_A) &= \text{penalty}(A) \cdot 10\% + \text{risk}(\text{penalties}_B) \\ &= 500\ € \cdot 10\% + 245\ € \Rightarrow \underline{295\ €} \end{aligned} \tag{8.23}$$

Recursively checking for job E, whether to initiate an FT-mechanism for it, has the result that the most profitable solution is to cancel the job. Since the first termination condition is fulfilled, no further iteration step would be performed. The resulting schedule from the chain of these FT-mechanisms is depicted in Figure 8.3.

The migration of job A to one of the resources used by B and the migration and time-shift of job B will be only performed if the costs for the initiation of all FT-mechanisms are below the summarised benefits benefit(FT_A).

- benefit(FT_B) $= 700 - 245 = 455$ €
- benefit(FT_A) $= 500 - 295 = 205$ €

This means that the total cost for the migration of A and B including the cost for the technology development and the data transfer must not be below 205 €. Otherwise, this chain of FT-mechanisms is not profitable. Note that dependent on the earliest start time of job D, the scheduler will execute this job on resources r_5 or r_6. The re-integration of the schedule will be performed by backfilling. In the case the earliest start time of D is after 11:30am, it cannot be executed before 1:30pm and the only possibility to prevent an SLA violation is that the resource r_1 would be re-available before 2pm.

The same evaluation process performed for job B has to be performed for job C. However, the deadline for C equals the planned end time and therefore no time is available for a migration. Consequently, a recursive analysis does not have to be performed for C. The decision whether to initiate a chain of FT-mechanisms depends on the cost for performing the FT-mechanisms for A and B as presented previously. If this chain is not valid under consideration of benefit and cost, the results of Section 8.3.3.1 are applied, i. e. performing no FT-mechanism for A.

This example computation showed the mechanism of evaluating FT-mechanisms according to the concept and cost definition of Section 8.3.2.1. Furthermore the termination conditions defined in Section 8.3.2.2 are applied.

8.3.4 Runtime Analysis

The recursive analysis of initiating several FT-mechanisms might be complex and expensive if all jobs in the system have similar requirements, similar PoFs, and similar penalty fees. Before integrating recursive checks in the resource management, this complexity is essential to be known. This section presents the runtime analysis of performing recursive checks. In the best case, no recursive analysis is required if a free resource can be used to compensate for the resource outage to be handled. Accordingly, the focus is on the runtime in the worst case scenario. Determining the runtime requires a consideration of the maximum number of recursion levels (see Section 8.3.4.1) as well as the effort performed in each recursion level (see Section 8.3.4.2).

8.3.4.1 Maximum Number of Recursion Levels

According to the termination criteria described in Section 8.3.2.2, the cost c of performing an FT-mechanism must be taken into account. The risk of paying a penalty fee for a job j when applying one or a chain of FT-mechanisms for job j corresponds to

$$\mathrm{risk}(\mathrm{penalties}_{j|FT}) \quad = \quad m \ . \tag{8.24}$$

If j is the job affected by a resource outage, performing an FT-mechanism for j has impact on other jobs j_e which are either cancelled, migrated, or time-shifted. According to these effects, in Section 8.2.1 the risk estimations for job j have been defined. It can be written:

$$\begin{aligned} \text{risk}(\text{penalties}_{j|FT}) &= \text{penalty}(j) \cdot Pr(\text{SLA violation of } j) \\ &\quad + \sum_{j_e} (\text{penalty}(j_e) \cdot Pr(\text{SLA violation of } j_e)) \\ &= \text{penalty}(j) \cdot Pr(\text{SLA violation of } j) \\ &\quad + \sum_{j_e} \text{risk}(\text{penalties}_{j_e}) \end{aligned} \tag{8.25}$$

If $m \geq \text{penalty}(j)$, the recursive analysis stops since the initiation of the FT-mechanisms is less profitable than paying the penalty fee of j. Replacing the variable m with the penalty and risk term in Equation 8.25 and using the stop condition leads to

$$m = m \cdot Pr(\text{SLA violation of } j) + \sum_{j_e} \text{risk}(\text{penalties}_{j_e}) \tag{8.26}$$

The number of recursion levels depends on the jobs j_e affected by the FT-mechanism for job j. In order to determine the maximum of recursion levels, the number of jobs j_e have to be determined having a higher total risk than the penalty m of job j. Equation 8.26 can be reformed to

$$\sum_{j_e} \text{risk}(\text{penalties}_{j_e}) \geq m\,(1 - Pr(\text{SLA violation of } j)) \tag{8.27}$$

One termination condition ensures that the benefit of performing an FT-mechanism is higher than the cost c which results from performing the FT-mechanism, i. e. benefit $> c$. The benefit was defined as

$$\begin{aligned} \text{benefit}_{j_e} &= \text{risk}(\text{penalties}_{j_e|FT}) - \text{risk}(\text{penalties}_{j_e|\neg FT}) \\ \Longrightarrow^{\text{benefit}>c} \text{risk}(\text{penalties}_{j_e|FT}) &\geq \text{risk}(\text{penalties}_{j_e|\neg FT}) + c \\ \Longrightarrow^{\epsilon>0} \text{risk}(\text{penalties}_{j_e|FT}) &\geq \epsilon + c \end{aligned} \tag{8.28}$$

$$\tag{8.29}$$

If performing the recursive analysis, FT-mechanisms are performed for jobs j_e, hence it follows from Equations 8.31 and 8.28:

$$\sum_{j_e} \epsilon + c \geq m\,(1 - Pr(\text{SLA violation of } j)) \tag{8.30}$$

Resulting, with p as the probability of an SLA violation of j, the maximum number of recursion levels is

$$\frac{m \cdot (1-p)}{\epsilon + c}\,. \tag{8.31}$$

8.3.4.2 Total Operating Cost

The number of evaluations in a recursion level depends on the jobs in the schedule. Since only one single job is executed on a resource, the maximum number of jobs running on the same time is limited by the number of compute nodes of the cluster. In the worst case only single jobs are executed in parallel on the cluster and consequently as many jobs are scheduled as nodes are assigned to the cluster. Let the cluster consists of n nodes and the job affected by a resource outage is job j_n. As a consequence, if searching for an alternative resource which can be used from job j, at most $n-1$ comparisons are made. In the case none of the jobs scheduled on the same time as j_n is a best-effort job or can be time-shifted without an SLA violation, the recursive analysis evaluates whether an FT-mechanism can be performed for those jobs $j_i, i < n$. In the worst case, all jobs $j_i, i < n$ have a lower penalty fee than j_n and consequently a recursive analysis is performed for each job j_i. In the next recursion level for each job j_i, n comparisons are made. To generalise, in the recursion level i, n comparisons are made $\Rightarrow n^i$ operations in total.

In order to get a better performance, a heuristic can be applied which performs first of all the complete evaluation for that job j_i with the lowest penalty fee. In the case that initiating a chain of FT-mechanisms for job j_i is not cheaper than violating the SLA of job j, the evaluation is performed for job j_k with the second lowest penalty fee. Following this strategy might be faster, but in the worst case this heuristic might not reduce the total number of operations.

Each operation includes the estimation of the probability of an SLA violation for two jobs: the probability for the job which might be executed on that resource after performing an FT-mechanism and the probability for the job which is affected by this FT-mechanism. The operation cost for estimating the PoFs and performing scheduling checks and modifications are constant for every job and is not considered in the following.

Combining the maximum number of recursion level before a termination might be performed with the number of comparisons of the operations leads to a complexity of:

$$n^{\frac{m\cdot(1-p)}{\epsilon+c}} \tag{8.32}$$

$$\Longrightarrow \mathcal{O}\left(n^{m\cdot(1-p)}\right) \tag{8.33}$$

where n is the maximum number of jobs running in parallel, m is the penalty fee of the job j affected by a resource outage, and p is the probability of an SLA violation of job j. The probability p is known after evaluating which resource can be used in scope of an FT-mechanism for job j. In the case that resources are similar stable and probabilities of failures do not significantly differ for a job when using different resources, p corresponds to the probability estimated during SLA negotiation.

8.4 Reacting After Resource Failures

The previous sections followed the assumptions that the evaluation of FT-mechanisms has been initiated since an unstable resource state has been monitored. The identification of an

unstable resource state is however not always possible and the RMS has to react to resource failures. The resource outage of a compute node results in the availability of less resources which implies that often not all SLA bound jobs can be executed. A risk aware management of resource failures is required in order to ensure that the most profitable solution will be executed. When the scheduler is handling resource failures, it is acting in scope of a targeted Grid Risk Management process, since the outage implies an SLA violation of one specific job if no countermeasure is initiated. The scheduler has to evaluate whether FT-mechanisms planned during the SLA negotiation are able to prevent an SLA violation. It also has to consider the priorities of jobs in the schedule in order to initiate the in expectation most profitable solution.

Consider a single or parallel job j with $r \in s(j), s(j) = r_i | t_s - t_e$ and resource r fails. If j is an SLA bound job, the scheduler is notified about the resource outage and initiates a targeted Grid Risk Management process. If j is performed in scope of a best-effort service, it is rescheduled without applying Risk Management. Accordingly, best-effort jobs are left out of consideration since for those no guarantees have to be met.

Either the resource failure of r can be compensated by free resources in the cluster, by an FT-mechanism planned during the SLA negotiation, or a complete new schedule has to be generated. It is necessary to generate a complete new schedule in order to define the resource allocation according to current job priorities resulting from the previous schedule and system state. Note that the described strategies have been defined under the assumption that free resources are rare because of a high system utilisation. No restrictions are made concerning the job state, i. e. either j is running or has not started yet.

The description of the Risk Management in the post-negotiation phase focuses on only one resource outage, however it can handle simultaneously occurred resource failures in the same manner. If resources fail at the same time which have been used by different jobs, for each job the evaluation of using a planned FT-mechanism is performed as described in Section 8.4.1. Job requests for alternative resources are handled in a first come first serve order. If not enough alternative resources are available to compensate for resource failures of all SLA bound jobs, a complete new schedule is generated (see Section 8.4.2).

8.4.1 Using Planned FT-Mechanisms

Section 8.1.3 described the FT-mechanisms which may be planned during the SLA negotiation for a job j. In this section the usage of these mechanisms in the case of a resource failure is considered. When reacting on an unstable monitored resource state, the scheduler is able to plan these FT-mechanisms even after the SLA negotiation. For example checkpointing can be initiated to handle unstable resource states or the checkpointing frequency can be modified. These opportunities are not provided after one resource used has already crashed. Thus, a strict strategy can be followed when managing a resource failure. Finally managing a resource failure has the highest priority since no reaction of the RMS will lead to an SLA violation. Considering resource outages for running jobs not all FT-mechanisms are appropriate to prevent an SLA violation.

Since resource r has crashed, checkpointing cannot be initiated after detecting this threat of an SLA violation. If checkpointing has not been initiated from the beginning, j has to be

rescheduled with the duration $t_e - t_s$, which either conforms to the job duration specified within the SLA or to an extended one defined by the scheduler during the SLA negotiation. If j was running and no checkpointing was initiated, preventing an SLA violation is a challenge for parallel jobs having hard time constraints. The difficulty for a successful prevention results from the fact that the complete job has to be executed from the initial state. For parallel jobs already running several hours before the resource outage, it implies that in most cases the reservation duration cannot be adjusted accordingly and a complete rescheduling is necessary (as described in Section 8.4.2). In order to increase the probability of restarting a parallel job in the given reservation slot, the job duration can be extended (see Section 7.4). If the job duration has been extended during the SLA negotiation, reservation slot $[t_s, t_e]$ is longer than the execution time defined in the SLA. Consequently, opportunities are provided to restart the job without modifying the reservation duration on the other $r_i \in s(j)$. The restart calls for that the affected sub-job of j is mapped to an alternative resource for the same time interval $[t_n, t_e]$ where $t_n = \max\{t_s, \text{now}\}$. The success of finding an alternative resource is influenced by other FT-mechanisms planned during the SLA negotiation.

Spare resources supports to find an alternative resource after a resource outage. These spare resources are either reserved for one specific job or assigned to a pool of spare resources. If job j has an FT-reservation and a resource outage occurred during its job execution, one of the dedicated spare resources is used as an alternative resource. The number of additionally reserved spare resources is then decremented since one resource outage has been compensated. The compensation results in decreasing the number of spare resources since the estimated PoF has considered the probability of a resource outage. Thus, not reserving an additional node conforms to the risk reduction plan of the SLA negotiation. In the case j has not started yet, the number of dedicated spare resources are not lowered since the failure of r before the job execution is not reflected in the PoF estimated.

If either j is planned, all dedicated spare nodes have been used before, or no FT-reservation was made, j has to search for another alternative resource to compensate for the outage of r. If the system utilisation is less than 100% and free suitable compute resources are available, j uses on of these. In the case no free resource is available and j is an SLA bound job, j may use a resource of the pool of spare resources. If all resources of the spare pool have been used by other jobs to compensate for their resource failures, a completely new schedule has to be created (as described in Section 8.4.2) since the scheduling has to be performed according to current execution states.

Section 7.4 pointed out that, in addition to these FT-mechanisms, a redundant job execution might be planned during the SLA negotiation. Since the redundant job execution was planned as a means to reduce the feasible PoF, the same strategy should be followed as for the usage of dedicated spare resources. If j has already started, this resource and execution failure has been considered when estimating the PoF during the SLA negotiation. The failed job instance will be only replaced by a new one if the system utilisation is so low that the failed instance of j can be scheduled without using the pool of spare resources. If j has not been started, the instance which was using resource r may also use the pool of spare resources since this failure has not been considered in the PoF estimation.

The FT-mechanisms planned during the SLA negotiation provide opportunities to prevent an SLA violation after a resource failure. In particular checkpoints are important since the

scheduling to resume the job execution can be the easier fulfilled the lower the remaining job duration is. Checkpointing combined with the extension of the job duration is even better for parallel jobs. In order to apply any FT-mechanism, in any case an alternative resource has to be found to resume or restart the job. FT-reservations planned during the SLA negotiations increases the success probability of finding an alternative resource. If those have not been made during the SLA negotiation and not unbounded resources are available because of a high system utilisation, SLA bound jobs use resources of the spare pool in order to compensate for resource failures. Dependent on the job state, using such specific alternative resources – either dedicated ones or assigned to the pool – is allowed or has different effects. The reason for a differentiation according to the job state is justified since the resource outages during the job execution are considered within the PoF, the failures before the job execution however are not considered. If not enough alternative resources can be found, a complete new schedule is generated as described in the following section.

8.4.2 Generate New Schedule

Compensating a resource failure of an SLA bound job is not always possible if the system utilisation is high and several resource outages have occurred, i. e. no free resource is in the pool of spare resources. In such a situation the simple strategy could be followed to outsource those jobs which could not be inserted again in the schedule. However, outsourcing an SLA bound job has to disadvantages: firstly, the provider gives away the control of the job execution; secondly, the provider has to charge the other provider for the job execution. Since jobs requesting for alternative resources are handled in the *First Come First Serve* (FCFS) order, the job to be outsource could have strict requirements regarding PoF, resource capacities, etc. Hence, outsourcing always the job which could not find an alternative resource is in expectation not the most profitable solution for the provider.

To modify the job execution according to realise in expectation the most profitable solution, a complete reschedule is performed. Jobs are prioritised according to the expected profit earned from a job execution as well as their expected loss.

Definition 8.4.1 (Expected Profit)
The expected profit for performing a job j is defined as:

$$\text{expectedProfit} \quad = \quad (\text{reward} \cdot (1 - PoF) - \text{remainingCost}) \tag{8.34}$$

It considers the profit which is made if the SLA is fulfilled in combination to the success probability, i. e. (1-PoF).

The remaining cost for the job execution is the product of the number of nodes used from resource j, the remaining execution time, as well as the internal cost for using a CPU per hour. It is important to remark that the internal cost have to be considered which will differ from the cost for one CPU-hour paid by end-users. The price for a CPU-hour will be defined according to market mechanisms and might change over time.

Definition 8.4.2 (Expected Loss)
The expected loss for a job j is considering that the job will not be scheduled and its SLA will be violated. Accordingly, it does not take into account the PoF estimated and only reflects

the cost which occurs in the case of an SLA violation:

$$\text{expectedLoss} = \text{spentCost} + \text{penalty fee} - \text{remainingCost} \tag{8.35}$$

The spent cost results from the resource usage of j until now. The remaining cost is defined through the remaining execution time and the requested number of nodes determined in the SLA. The remaining cost are subtracted from the expected loss since these cost does not incur, if the job is not scheduled again. Listing 8.1 details the calculation of these values. Equation (8.35) is defined according to the assumption that the revenue is paid to the provider independent from its success of providing the requested service. In the case that the provider earns only the revenue if it has fulfilled the SLA, the revenue has to be added to the expectedLoss.

Listing 8.1: Calculating Expected Loss of an SLA Violation

```
static double estimateExpectedLoss
{
  time_t now;
  now = time(NULL);

  ulong performedExecutionTime = now - scheduledStartTime + performedExeTime;
  if (performedExecutionTime < 0)
    performedExecutionTime = 0;

  ulong spentCost = numberOfNodes · cpuHourCost · performedExecutionTime;

    ulong remainingExecutionTime = duration - performedExeTime;

    if(job is allocated)
    remainingExecutionTime = (scheduledStartTime + duration) - now -
        performedExeTime;

  double remainingCost = numberOfNodes · cpuHourCost · remainingExecutionTime;

  double value = (spentCost  + penalty fee - remainingCost);

  return value;
}
```

The job priority is defined through expected profit - expected loss in order to prefer those jobs which will result in more profit. It is important that the *expected* profit is considered and not only the profit, i. e. revenue - internal cost for the job execution, since each job has a specific PoF which was estimated during the SLA negotiation. To apply Risk Management it is necessary to take this PoF estimation into account. Finally if two jobs have the similar expected profits, it is meaningful to prefer the execution of a job having a lower PoF than a job having a higher PoF. Since the rescheduling has been initiated because not all SLA bound jobs can be executed without an SLA violation, not all jobs will be able to be inserted in the new generated schedule. Those jobs which could not be scheduled are in expectation at least profitable to execute. Either the risk of an SLA violation is accepted or these jobs are outsourced. The prioritisation used for the scheduling has advantages for the outsourcing. The in expectation least profitable jobs will have low resource requirements or PoF constraints in comparison to other jobs of the schedule. Thus, the provider has to pay less for outsourcing

such a job than a job with strict requirements. Furthermore, the most profitable solution was found independent from the success of an SLA negotiation with other providers.

If historical activities have shown that in most cases appropriate SLAs could be agreed for outsourcing, the assumption can be made that jobs will be able to outsource if not executable on the own resources. As a consequence, the cost for outsourcing should be taken into account when estimating the remaining cost. The rewards requested by other providers can be hardly estimated since these might depend on policies, system utilisation, etc. However, the cost for transferring data to the other provider can be estimated independent from the SLA offers from other providers. The data transfer is of crucial importance in scope of a migration since checkpoints of parallel jobs running on many compute nodes might have a size of several gigabytes.

The new generated schedule is build by inserting jobs one after another into the schedule according to an increased priority. As a result of this complete new schedule generation, jobs which have run may be paused and resumed later in the schedule. In order to avoid the computation steps performed for such jobs, a checkpoint should be generated before allocating the new job. System administrators may define a prioritisation factor of jobs which should be executed at the time point of evaluation. This factor is then adjusted during the risk review, if necessary. The reason for such a prioritisation factor is that preferring currently scheduled jobs[2] ensures that the new schedule is not completely different. A complete different schedule hold the danger that many jobs which have been scheduled before cannot be inserted in the schedule. However, the danger concerns only jobs whose start time is close to the evaluation time. A relatively good quality is ensured, since before the new generation of the schedule, a valid schedule had exist. It is important to find an accurate prioritisation factor in order to stick to the Risk Management strategy. If using a scheduler developing optimal solutions according to a priority (here the expected profit - loss), such a factor need not to be considered.

In order to avoid that arbitrary jobs are outsourced if these do not find enough alternative resources after a resource failure, a new schedule has to be generated. The schedule generation is based on an increasing priority defined through the expected profit and loss for each job. Taking into account the PoF when determining the expected profit is crucial since the PoFs of jobs might significantly vary. The jobs which could not be inserted in the new schedule will be outsourced if an appropriate SLA can be agreed. If those SLA negotiations are not successful, the provider implement by the new schedule the solution which is in expectation the most profitable one. Note that for generating the new schedule optimal mechanisms can be used, however, in the Grid resource management the computation of the schedule has to be performed in several milliseconds.

[2] either running or they have run and have been affected by a resource outage

8.5 Recapitulation of Risk Management in Post-Negotiation Phase

As identified in the requirement analysis the most threatening event for an SLA violation is a resource outage. In order to handle resource failures in the post-negotiation phase, FT-mechanisms are of crucial importance. The Risk Management performed during the negotiation phase may be useful in order to prevent SLA violations in the case of a resource failure. In particular, an initiation of checkpointing and an extension of the job duration increases the probability of preventing an SLA violation. A challenge in a system with high workload is to find an alternative resource. At this stage the application of Risk Management is necessary in order to implement the most profitable schedule.

The evaluation of initiating an FT-mechanism is performed in two different scenarios: an evaluation can be initiated either because of an unstable resource state (see Section 8.2 and Section 8.3) or after a resource outage has affected an SLA bound job (see Section 8.4). The main difference of both scenarios is that an immediately reaction is not necessary if an unstable resource state has been monitored. However, after a resource failure a strict strategy has to be followed since for evaluating and comparing different plans is not possible.

Expect checkpointing, the FT-mechanisms which can be applied in scope of the Risk Management are the same in both scenarios. Figure 8.4 depicts the Risk Management activities in the post-negotiation phase. The initiation of any FT-mechanism for an affected job j is performed in scope of risk reduction. In scope of an unstable resource state, the FT-mechanism or the chain of FT-mechanisms are only initiated if the total risk is lowered. After an occurred resource outage the probability of an SLA violation is 100% for the affected job and consequently its risk is reduced if any FT-mechanism is executed.

Risk acceptance corresponds to accept the SLA violation of a job. This Risk Management strategy is followed if the initiation of an FT-mechanism for the job has an impact on jobs whose execution is more profitable in expectation. Instead of accepting the risk, transferring risk to another resource provider is possible if an appropriate SLA has been agreed for outsourcing.

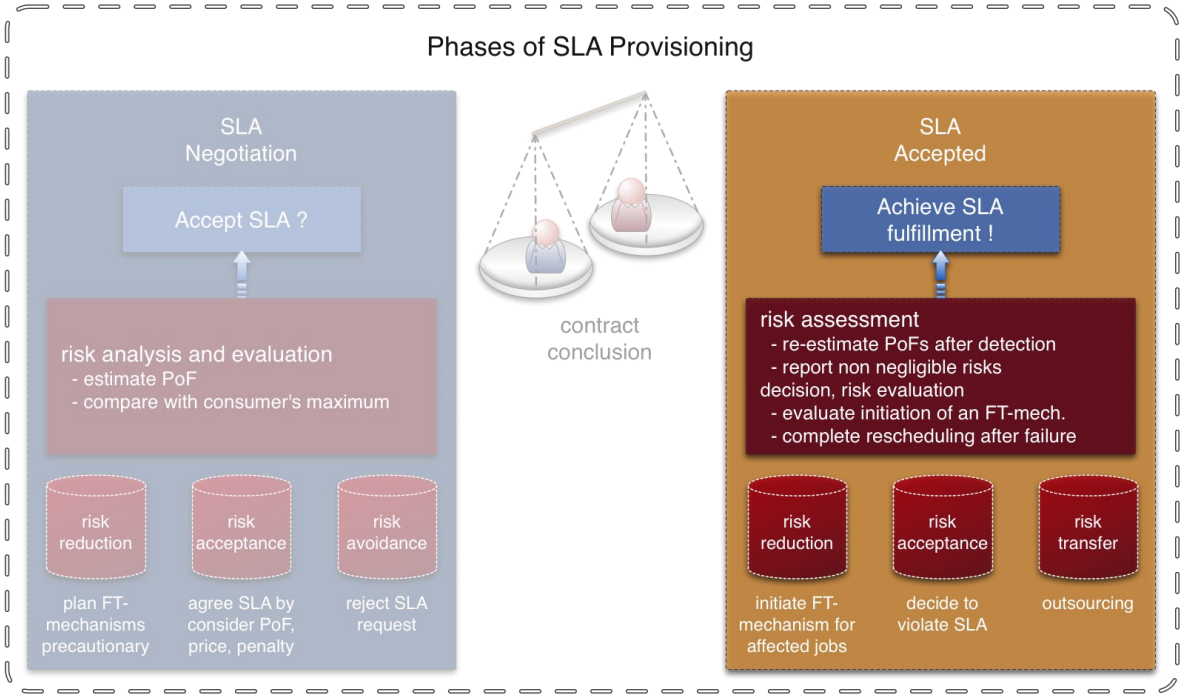


Figure 8.4: Addressed Risk Management In Post-Negotiation Phase

Chapter 9

Evaluation Results

The previous chapters presented Risk Management processes realisable in the Grid as well as Risk Management activities which can be initiated during the SLA negotiation and in the post-negotiation phase to support the provider's SLA provisioning. The benefit of using such Risk Management mechanisms is measurable by the provider's profit increase which can be traced to applying Risk Management. This chapter presents evaluation results in order to demonstrate the benefits of applying the Risk Management developed.

Before the evaluation results are presented, it is necessary to describe the underlying system (Section 9.1). The results of the risk assessment are shown in Section 9.2. The Risk Management related evaluation results have been produced by running jobs in a basic scenario which is described in Section 9.3. After showing the results in Section 9.4, an overview about other imaginable scenarios to test the Risk Management integration is given in Section 9.5. Section 9.6 completes the description of this work by contextualising the developments with techniques, objectives, and steps performed in Risk Managementin general and IT-Risk Managementin particular.

9.1 System Design

After presenting an overview of the AssessGrid project and its outcomes in Section 9.1.1, the risk aware concept of the SLA negotiation is detailed in Section 9.1.2. Since this work focuses on the processes within the resource management, Section 9.1.3 describes the system in the Grid fabric.

9.1.1 AssessGrid

The AssessGrid project – Advanced Risk Assessment and Management for Trustable Grids – is funded by the European Commission under contract IST-031772 [AssessGr 08]. The overall objective of the project is to integrate risk assessment and management into all Grid layers in order to close the gap between SLAs as a concept and an accepted tool within a commercial Grid environment.

The integration of risk assessment and management is realised for the provider, broker, and end-user by modifying their processes to consider *Probabilities of Failures* (PoFs) of resources and SLAs. SLAs are based on the WS-Agreement specification [Andr 07] and enhanced by a

parameter for the PoF for the SLA. This value is part of the SLA negotiation and acts as an decision criterion during the SLA negotiation and in the post-negotiation phase.

The project AssessGrid aims to satisfy the demands for transparent and understandable risk evaluation by extending the Grid technology with methods for risk assessment and management as core services of future Grids. Since end-users, brokers, and providers have different perspectives on risk enhanced Grid services, they define the three major AssessGrid objectives: the end-user seeks a reliable and trustworthy provider, the broker looks for the best offer for its customer, and the provider aims to reduce the risk of SLA violation. Furthermore, providers need objective measures to lower the execution risk and to analyse their infrastructure in order to remove bottlenecks. The research and development work consists of three phases reflecting the three addressed scenarios. Every phase ends with a defined outcome which will enrich the Grid with additional components and information and is ready for demonstration and review.

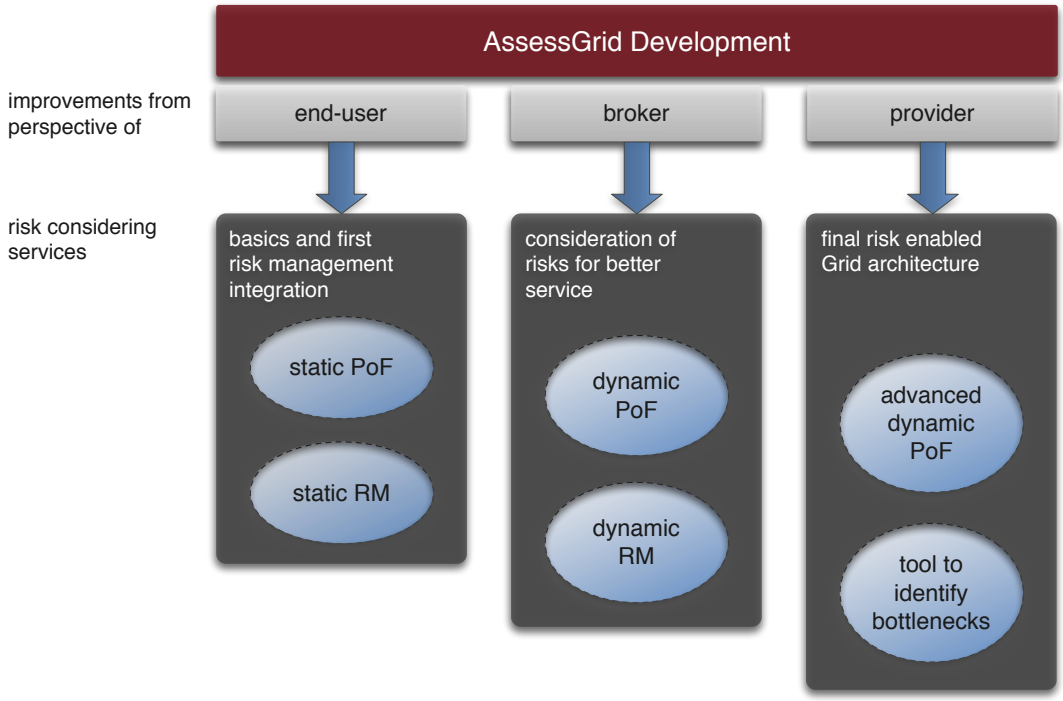


Figure 9.1: Outcomes of the AssessGrid Project

The AssessGrid project generates three outcomes as depicted in Figure 9.1. The software is completely open source and implements a risk aware SLA provisioning for providers and brokers. In addition to this the end-user's interface enables them to see the AssessGrid specific SLAs which include the PoF estimation as an additional parameter.

1. **Risk Aware End-user Client:** The first outcome focuses on the end-user perspective. It contains basic mechanisms on Risk Management considering solely static data for the risk assessment. In this outcome basic Risk Management activities are integrated into all Grid layers as a first step to implement risk awareness. The first prototype realises a risk aware SLA negotiation by supporting the modified SLA structure and

using a risk aware resource allocation in the Grid fabric. The prototype was deliverable in October 2007 [Voss 07e].

2. **Risk-enhanced Broker Service:** In the second phase the AssessGrid system is enhanced to consider Risk Management methods especially for improving the quality of brokers' service and for provisioning of workflow jobs. The risk assessment will be dynamic. On the Grid provider layer this outcome realises Risk Management activities in the post-negotiation phase in order to support the provider in managing resource failures. The second outcome will be released in May/June 2008.
3. **AssessGrid Risk Management System:** This third prototype will be the main AssessGrid outcome, a vertically integrated solution including all planning, monitoring, and Risk Management methods for the Grid end-user client, Grid broker, and Grid provider. It further enhances the risk assessment by considering statistical/historical data, ad-hoc input, and general data. Moreover, this solution will offer tools for a simplified system management and for confide in providers' offers. These tools are very beneficial for the competitiveness of providers. Overall, this outcome will enhance the second outcome from the provider's perspective. It will be released at the end of the project in December 2008.

The results of this thesis are used within the AssessGrid project and realise the Risk Management processes in the Grid fabric. In order to describe the Grid environment of these developments, Section 9.1.1.1 follows with a summary of the risk awareness in the broker and end-user layer. Economic aspects which arise from the risk awareness for end-users, brokers, and providers can be found in [Voss 07d].

9.1.1.1 Broker and End-user Layers

In AssessGrid the Grid broker has specific risk aware tasks and services which differs the risk aware broker from non risk aware brokers which are only responsible for assigning jobs to several Grid providers. risk aware brokers in particular benefit from publishing PoFs within SLAs in two cases. First of all, the Grid broker is able to improve the provisioning of workflow jobs since sub-jobs having a high PoF can be executed by different resource providers in parallel. Such an approach conforms to the FT-mechanisms executable in the Grid fabric, are however implemented on a different abstraction level. Secondly, the Grid broker benefits from the high number of SLA negotiation it is involved in. Evaluating the ratio of SLA violations in comparison to the published PoF might be an indication whether the provider is honestly publishing PoFs or lying in order to accept an SLA. [Gour 08] presents a reliability measure in order to classify providers as reliable, moderate, or unreliable according to the PoF offering and SLA violations observed. The reliability measure is used to adjust a PoF in an SLA when the broker is involved in the SLA negotiation. Additionally, end-users can asks for a PoF adjustment by sending a request to the broker's confidence service, if they are directly negotiating with a provider.

Risk awareness in the Grid end-user layer is reflected in the portal used for SLA negotiation and job submission. The portal implements the AssessGrid's specific WS-Agreement concept by offering the end-user to define and evaluate PoF estimations. The maximum PoF which they are willing to accept can be inserted in the SLA and the PoFs of SLA offers are visualised by

a traffic-light system. Furthermore the portal enables the end-user to negotiate with arbitrary Grid providers and brokers. Risk aware specific features such as the confidence service of the broker can be called on demand.

9.1.2 SLA Negotiation

SLAs are realised by implementing the WS-Agreement specification (see Section 3.3). Implementation details can be found in [Batt 07]. A provider does not differ if it negotiates with a broker or an end-user. Merely, policies might lead to a different behaviour in the offer generation and SLA acceptance. The WS-Agreement specification defines that an SLA is sent from the agreement initiator to the agreement provider. The agreement provider is able to accept or reject the SLA. Since a Grid service provider is usually not able to generate SLAs suitable for specific job executions without having detailed knowledge about the end-users, the agreement initiator is always the service consumer. If a broker is acting as an intermediate negotiation actor [Djem 06], it is the agreement initiator and the service consumer is specified through the agreement owner.

In order to integrate the PoF as an additional parameter within the SLA, an AssessGrid namespace has been defined. Furthermore, the workflow of an SLA negotiation has been slightly modified since agreement initiators may perform a previous check which providers are willing to accept an SLA and what is their revenue. Consequently, the agreement initiator is sending an SLA request to the provider in which the service terms and guarantee terms, as well as the maximum accepted PoF and the penalty fee are defined. The Grid service provider evaluates the risk of agreeing this SLA by making a risk aware reservation. Based on this risk aware reservation, the feasible PoF is known in addition to the necessity of initiating risk reduction plans in order to fulfill the maximum accepted PoF. The provider decides based on this information whether it is willing to accept the SLA and defines the revenue. In addition a provider might publish its estimated PoF as described in Section 7.1. Since the provider is modifying in this stage the WS-Agreement, the implementation would not follow the specification if the Agreement Initiator will accept or reject the SLA from the Grid service provider. Consequently, the internal check in the provider as well as the definition of the PoF and the revenue are performed in a tentative manner – using a `getQuote` function as depicted in Figure 9.2. After the provider has evaluated the feasibility of the SLA, it deletes the associated job reservation. Based on tentative SLAs received from several providers, the agreement initiator is able to select one of these and modify their SLA according to the provider's data. This SLA is then sent to the provider by using the `createAgreement` method. The provider evaluates again whether and how the SLA bound job can be executed and decides to accept or reject the SLA. Note that the risk aware reservation workflow as described in Chapter 7 is executed after receiving a `getQuote` or `createAgreement`. The difference is that after handling a `getQuote` request, the tentative reservation is deleted. Note that before sending a `getQuote` request, an agreement initiator asks agreement providers for their SLA template which describes the resources and services offered by the provider.

The WS-Agreement specification is very powerful regarding the definition of *Service Level Objectives* (SLOs). For each SLO a penalty fee might be defined in order to compensate for the loss of the agreement initiator. Since this is quite complex, a simplified approach is used. In the SLA only the revenue, which is paid from the agreement initiator to the provider, and the

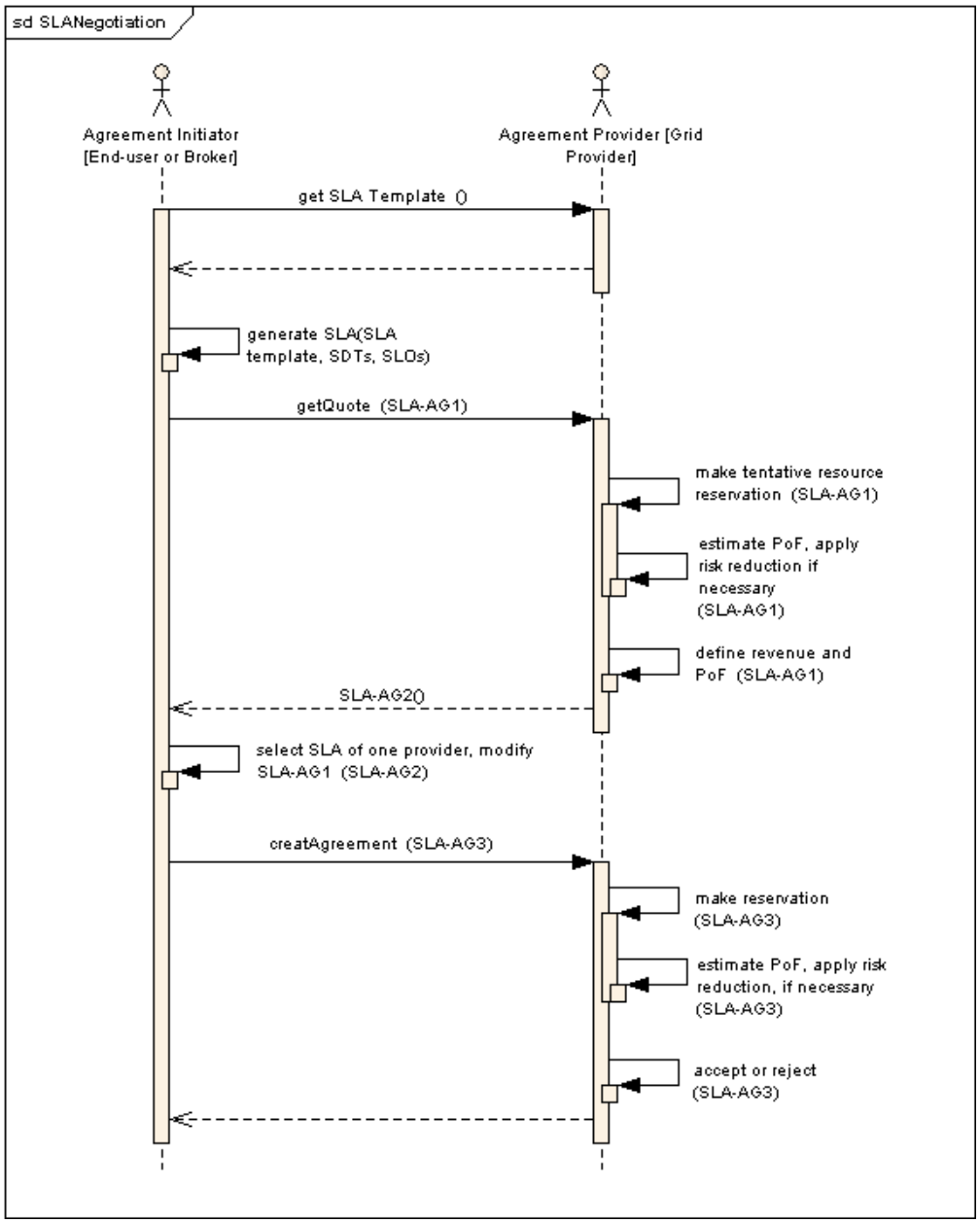


Figure 9.2: Workflow of SLA Negotiation between Initiator and Provider

penalty fee, which is paid from the provider to the end-user, are defined. If the provider could not fulfill any SLO as negotiated, it pays the associated penalty fee to the Agreement Initiator. SLOs are assigned either to the Agreement Owner or to the Agreement Provider, i. e. either it has to be fulfilled by the end-user or by the provider. For example an SLO in the responsibility of the end-user is that the input data of the job has to be available at the defined location until a specific time. If the data is not available on-time, the provider could not start the job execution as planned and consequently it is not responsible for the SLA violation. Under consideration of such a scenario, several rules have been defined. First of all, the penalty

paid by the Agreement Owner conforms to the reward defined in the SLA. Secondly, the first detected violation of an SLA is considered as the cause for an SLA violation and determines who has to pay a penalty – either the agreement owner or agreement provider. Thirdly if an SLA is violated, the revenue is paid to the provider and the provider pays the penalty fee to the consumer. According to this definition it is reasonable if the penalty fee is higher than the revenue in order to compensate for the loss of the service consumer.

Identifying the cause of an SLA violation might be a challenge since it can be a hardware or software failure. The provider is not liable for software failures since often the applications executed are implemented by the service consumers themselves. In addition the agreement owner has not fulfil their obligations if they have underestimated the runtime of the job and because of this underestimation the expected results have not been produced. In contrast to these reasons, the responsibility of an SLA violation is assigned to the Grid service provider, if it did not allocate the resources on-time or any part of the resources failed and it could not be resolved through FT-mechanisms. It may be that no resource failure is detected since the job failure has different causes, then the simplifying assumption is that this is the end-user's fault.

9.1.3 Resource Management

In order to implement Grid Risk Management processes within the Grid fabric, risk awareness has to be integrated in the provider's resource management. In this work the assumption is made that the *Resource Management System* (RMS) is planning based system. This section gives a summary about the RMS extended with Risk Management.

The *Cluster Computing Center* (OpenCCS) [OpenCCS 08] is a planning based RMS supporting advance reservations. OpenCCS has been enhanced to be useable within Grids by developing an interface to Globus Toolkit [Globus 08]. Furthermore it supports SLA negotiation according to the WS-Agreement specification as presented in the previous section. OpenCCS can use different scheduling strategies from first-come-first-serve, to shortest-job-next, up to SLA aware scheduling. Using the SLA scheduler, SLA bound jobs as well as best-effort jobs can be executed. For SLA bound jobs resource constraints as well as time constraints are ensured. To manage resource failures, several FT-mechanisms are integrated. Checkpointing and migration technologies have been integrated in OpenCCS in scope of the HPC4U project [HPC4U 08]. Rescheduling jobs is performed automated and outsourcing can be initiated if SLA bound jobs cannot be inserted in the schedule.

The scheduling in OpenCCS is performed from two modules - the *Planning Manager* (PM) and the *Machine Manager* (MM). The PM generates a resource-independent schedule whereas the MM assigns jobs to specific resources. The separation between PM and MM is based on the consideration of static and dynamic data about resource availability. The MM knows which resources are available and which are offline, whereas the PM just consider the number of resources when generating a schedule. The resource mapping of a job to a specific resource is modified by the MM if resources are not available which have been planned to be used. This separation has also a significant benefit in scope of risk awareness: two different levels can be realised, which dependent on whether the resources' stability of nodes of one cluster are similar or not. If these differ, the MM generates the specific schedule on a resource-level

already during the SLA negotiation in order to take into account the PoF for a specific resource set. If resources of the same cluster have the same PoF, no explicit resource assignment has to be performed during the SLA negotiation.

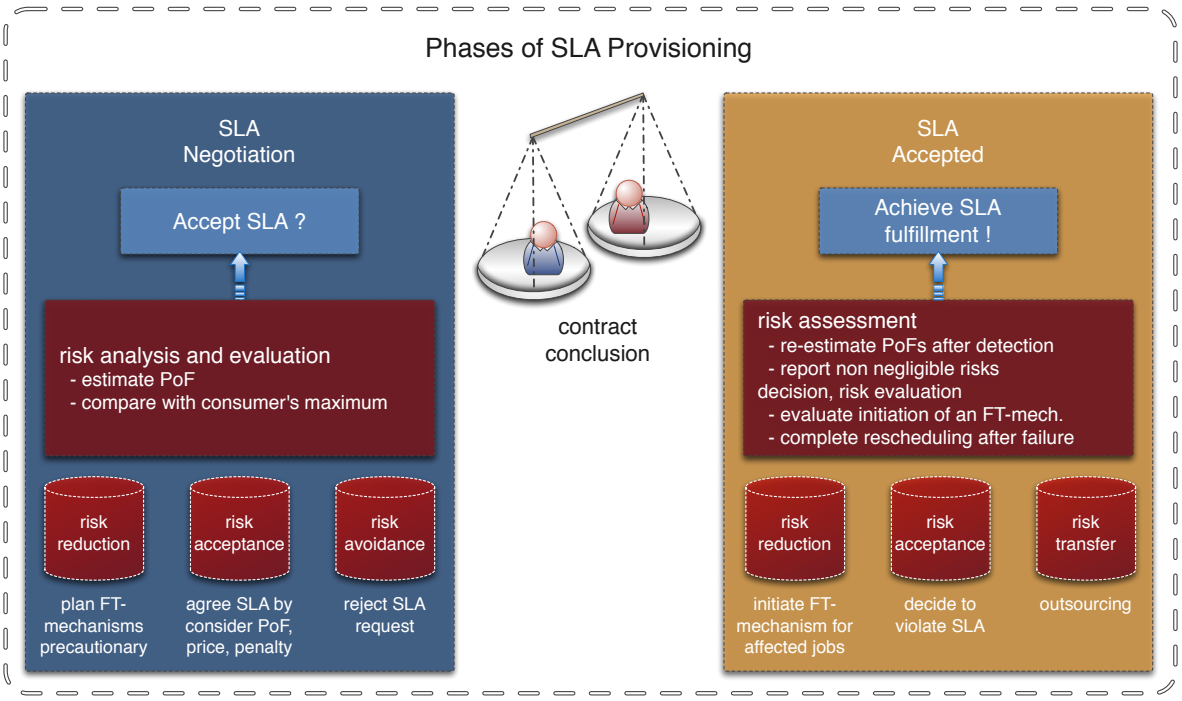


Figure 9.3: Risk Management for SLA Provisioning

The enhancement of OpenCCS in scope of risk awareness resulted in the development of an additional service responsible for the estimation of the PoF. The PoF estimations received of the risk assessment module are used in the SLA negotiation (as described in Chapter 7) as well as in the post-negotiation phase (as described in Chapter 8). Figure 9.3 repeats how Risk Management strategies are applied to support SLA provisioning.

9.2 Evaluation of Risk Assessment

In order to consider accurate failure rates of clusters, log traces from the *Los Alamos National Laboratory* (LANL) [CFDR 08] were used to evaluate the application of Risk Management in SLA provisioning. Section 10.2.2 details the analysis performed on monitoring data from LANL. The log file of the failures was used to determine PoFs according to the risk assessment model which was presented in Section 6.2 for jobs varying in their requested compute nodes and duration. Table 9.1 shows the estimated PoFs in comparison to the success of exemplarily runs. The exemplarily runs have been performed by randomly choosing 10.000 times a start time s for the job. According to these start times s_i the number of crashes in the log file were counted which would have affected the job starting at s_i. By determining the ratio of job runs without any node outage, the sampling converse probability and PoF were calculated. The

Number of Nodes	Duration	Pr(0 node fails)	PoF	Sampling PoF
1	1 Day	0.99426	0.00574	0.00511
1	10 Days	0.94405	0.05595	0.04400
10	1 Day	0.994405	0.05595	0.04711
10	10 Days	0.56226	0.43774	0.32704
128	1 Day	0.47853	0.52147	0.38446
128	10 Days	0.00063	0.99937	0.95865

Table 9.1: PoF Estimations and Sampling Observations

PoF estimation starts with determining the probability that no node crashes,i. e. the converse probability – *probability of Success* (PoS). Based on the PoS, the PoF is determined. Note that the listed PoFs do not consider any FT-mechanism, like the pool of spare resources or dedicated spare resources, and consequently they follow a conservative approach.

The PoFs are very similar to the sampling values. An equality can rarely be achieved since the figures are only probabilities. Furthermore, an estimation exactly of one percent is not possible. However, the small differences are accurate.

Table 9.1 only shows the probability that no node fails conforming to the probability of success if no FT-mechanism is initiated. In addition to these figures, it is interesting how many node crashes will probably occur during the job execution and have to be managed. Table 9.2 shows the probabilities that a job running for 10 days on 128 nodes is affected by exactly x crashes.

It is important to remark that these PoF estimations consider the job execution in a relatively stable Grid environment. Monitoring data of the Grid'5000 shows that resources are significantly less stable since the average uptime of a compute node is only 45 hours. Section 10.2.1 details an analysis of the stability of the resources in Grid'5000. Since this data has been used to develop the risk assessment model, in this chapter the comparison of the estimated PoFs with the monitoring data from LANL is presented. The validation of the PoF in context of the Grid'5000 can be found in [Voss 08b].

9.3 Basic Scenario and Parameters

The evaluation of the Risk Management is performed in scope of a basic scenario presented in this section. To clarify the types and requirements of jobs executed, Section 9.3.1 defines the assumptions made. In scope of SLA provisioning, the revenue and penalty fee are crucial. Their definition used in the basic evaluation scenario can be found in Section 9.3.2 and completes the information concerning job submission. In addition to the job definition, the FT-mechanisms supported from the RMS are crucial. As presented in Chapter 7 some FT-mechanisms can be used to lower the PoF during the SLA negotiation. Section 9.3.3 describes which FT-mechanisms are initiated as default and which may be initiated in the post-negotiation phase. The impact of the initiation of specific FT-mechanisms can be evaluated in more detailed scenarios as presented in Section 9.5.

x= # Node Crashes	Pr(x crashes during execution)	Sampling Probability
0	0.00063	0.04135
1	0.00464	0.07129
2	0.01711	0.11990
3	0.04204	0.12658
4	0.07747	0.11400
5	0.11421	0.10223
6	0.14030	0.08074
7	0.14774	0.05987
8	0.13612	0.04710
9	0.11148	0.04103
10	0.08217	0.02965
11	0.05506	0.01565
12	0.01917	0.01565
13	0.01917	0.01319
14	0.01009	0.01245
15	0.00496	0.00945
16	0.00228	0.00806
17	0.00000	0.00765

Table 9.2: Probabilities that x Nodes Crashes Affect a Job Running on 128 Nodes for 10 Days

9.3.1 Jobs

The basic scenario to evaluate the benefits of applying Risk Management in SLA provisioning focuses on parallel jobs for which a higher PoF exist than for single jobs. The higher PoF depends on the number of resources used as well as the lower flexibility of finding alternative resources. As described in Section 7.4, a parallel running job affected by a resource outage either needs an alternative resource immediately after the failure or has to be completely time-shifted. However, a rescheduling/resuming in a different time slot is often difficult if the system utilisation is high and the job requests for a high number of compute nodes. Due to their higher requirements and inflexibility, using Risk Management for parallel jobs is more important than for single jobs.

In the basic scenario the assumption is made that all jobs executed in the system are bound by an SLA. Hence, no best effort jobs are executed which is a valid assumption for a commercial Grid provider. As defined in the model of this work (see Chapter 6), the key definitions of the SLA are the number of nodes, the earliest start time, the deadline, as well as the job duration. Since SLAs have not been established yet, no job traces of SLA bound jobs are available. Consequently, a job trace of the Parallel Workload Archive [ParWorkl 08] was used in order to simulate a practical operation of the cluster. Further job traces can be found in the Grid Workload Archive [GWA 08].

The Parallel Workload Archive provides different job traces from various systems. In the basic scenario a job trace of the LANL was used since for this Grid also monitoring data about resource failures is available. The LANL's job trace [CM 5 Tra 96] available in the

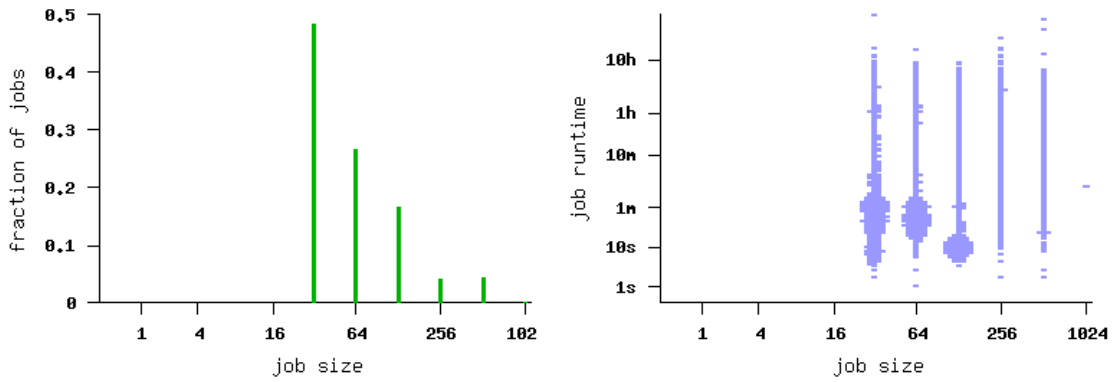


Figure 9.4: Distributions of Jobs in the LANL CM-5 Log [CM 5 Tra 96]

Parallel Workload Archive reflects the system utilisation of a 1024-node cluster. The logs about the resource failures have been used together with this job trace in order to simulate a Grid operation by a risk aware RMS. In order to be applicable, the time was scaled down of both traces, i. e. a job running one hour executed one minute in the simulation. Obviously, scaling the node crashes in this time interval was mandatory. This runtime adjustment has been also considered in the PoF calculation, i. e. the PoF for the original runtime has been estimated. Job sizes regarding number of nodes and runtime of the LANL's CM-5 log are depicted in Figure 9.4. Additionally, the cluster could only simulate to have 200 nodes and consequently, not all jobs listed in the job trace could be executed in the basic scenario.

The job trace defines several parameters according to the standard workload format about the job execution, however, the jobs were not bounded by SLAs and accordingly, few adjustments have been necessary. Note that the job trace reflects the usage of a queuing system and accordingly a job has got a waiting time until it was scheduled. The modification to run the job as an SLA bound job in a planning-based system are explained in the following.

The standard workload format lists a submit time of the job. This time is defined as the earliest execution time in the SLA. The requested CPU time in seconds is mapped to the job duration. The number of nodes for the job execution is defined in two values in the log file which might be inconsistent: first the requested number and the used number. In order to avoid inconsistencies the number of nodes requested in the SLA is defined as the minimum of both figures. Since the job trace describes the job execution performed, the exit code of the application. If the exit code equals to 0, the job was not successful which is defined as an SLA violation in the basic scenario. The log file contains only an end-time of the log but no time specification when the jobs was completed. Consequently, no end time could be mapped to a deadline. The deadline is defined as the submit time plus a multitude of the duration. This is necessary in order to transfer the job traces of a queuing based system into a planning based system. In job traces of the Parallel or Grid Workload Archive often only the submit time is defined and accordingly some additional time stamps have to be defined to run them as SLA bound jobs. In the simulations the buffer was a multitude of the duration, i. e. deadline = submit time $+ 5 \cdot$ duration, which resulted in a schedule with a high workload. The execution window might seems to be long, however without such a wide buffer, the system

utilisation was really low. For example when defining the length of the execution window as three times the duration, executing the same jobs leads to a schedule being 1.8 times longer. The generation of such SLA specific values has no effect on the evaluation results, these are out of scope and only applied to generate a high workload according to realistic job data.

9.3.2 Revenue and Penalty Fee

This work assumes that PoF is considered during the SLA negotiation and will decisively influence the definition of revenue and penalty fee since a more reliable service is more expensive. As stated in [Hass 07], PoFs should be reflected in the revenue: "*Another key parameter in determining price is risk. If the SLA has tight deadlines or high liability then the price should include an insurance premium to cover the liability.*" If risk/PoF is considered in the revenue, it is also considered in the penalty since both depend on each other. The assumption might be made that in an established Grid market, standard PoFs are defined of jobs by taking into account their resource and time constraints. According to a *standard PoF* (PoFS), the market price p is determined for using one CPU per hour.

For the evaluations the PoF was set according to the analysis shown in Section 9.2. The penalty definition depends on the buffer which determines the time the RMS has to compensate for resource failures. Hence it reflects the urgency/inflexibility of the job execution. This approach is similar to [Yeo 05].

$$\text{Buffer}(j) = \text{deadline} - \text{submit time} \tag{9.1}$$

$$\text{Let } p \quad \text{be the market price for using one CPU/hour}$$

$$\text{Penalty}(j) = \frac{\text{Buffer(j)}}{\text{duration}} \cdot \text{duration} \cdot \text{nodes} \cdot p \cdot \frac{(1 - \text{PoF})}{(1 - \text{PoFS})} \tag{9.2}$$

The buffer has to be determined according to the submit time/earliest start time and the deadline. If the buffer equals the duration, the provider has no time to perform any FT. Hence the probability is higher that an SLA violation cannot be prevented in the case of a resource outage. It may be reasonable to weight the ratio of the execution window and the duration less, i. e. the term $\frac{\text{Buffer(j)}}{\text{duration}}$ is multiplied with a factor $f < 0$. If execution windows are defined significantly longer, then such a modification should be applied in order to achieve a good ratio of revenue and penalty fee. Such questions will be autonomously clarified when the commercial Grid has been established. Finally such aspects depend significantly on behaviours and policies of individuals – service consumer and service provider.

Based on the resource requests and the PoF the provider defines the revenue as:

$$\text{Revenue}(j) = \text{duration} \cdot \text{nodes} \cdot p \cdot \frac{(1 - \text{PoF})}{(1 - \text{PoFS})} \tag{9.3}$$

The penalty fee is at least as high as the revenue. Since the provider is paid even in the case of an SLA violation, these definitions ensure that in the case of an SLA violation the service consumer really receives a contractual penalty. If the penalty fee would be lower than the revenue, the service consumer would have to pay for the service even if it was not provided as negotiated.

In other business fields the definition of contractual penalty fees is various. A violation of a conveyance contract are delays or flight cancellations which are caused by an event within the control of the airline itself. If then the airlines, such as Lufthansa, American Airlines, Emirates, are not able to get the customer to their final destination on the expected arrival date, they either refund them for the tickets or pay for accommodation and they can use the next possible flight. Not stated in the their conditions of carriage is that they are offering payments when flights are overbooked and not all passengers can be carried. Hence, service consumers either are refunded completely, i. e. the penalty fee equals the revenue the consumer has paid, or they receive an additional payment to compensate for the caused inconvenience, i. e. the penalty fee is higher than the revenue.

In the construction business it is common practice that service consumer pay not before the completed service has been delivered as requested. The liability for defects also often includes contractual penalties for each day which is delayed. These are important since service consumers have additional cost for alternative accommodation or have to extend the validity of other tenancy agreements. In this field of application the revenue is paid after service delivery. Since in the Grid the end-user has also to fulfill their obligations, it should be preferred that the revenue is always paid. If the SLA is violated, the provider pays a contractual penalty which is higher than the revenue. Comparing the ratio of revenue and penalty fee with the model in construction business, not fulfilling the SLA equals to a service delivery which is useless for the service consumer since it was delivered too late.

9.3.3 FT-Mechanisms

To show the full potential of the utilisation of Risk Management in SLA provisioning, all supported FT-mechanisms of the RMS OpenCCS have been activated. The usage of FT-mechanisms starts in the SLA negotiation by the initiation of checkpointing for each SLA bound job. This was generally activated and independent from the PoF. In addition to checkpointing, rescheduling is performed, if a job has not started yet, and migration to resume a job after a resource outage. Migration might be performed only internally on the same cluster or in scope of outsourcing. An internal migration might conform to involve only one new resource and continue the job on the other resources which have been used before and did not crash. In the scope of outsourcing, it usually makes no sense to migrate only one sub job since the connectivity between different clusters and different Grid sites is significant worse than within a cluster (see Section 6.1.1). As a consequence, if external resources have to be used, in this scenario always the complete job is migrated there.

The pool of spare resources was defined to be 5% of the complete cluster in order to be able to compensate for a few resource outages. Dedicated spare resources are reserved if the maximum PoF cannot be achieved otherwise. The upper bound of the number of dedicated spare resources which may be additionally reserved conform to 10% of the nodes requested by the job. By using this upper bound, a minimum profit margin is assured.

9.4 Evaluation of Applying Risk Management

The cluster was built of 200 nodes, from which 8, i. e. 5%, have been assigned to the spare pool. The simulation reflects the operation of the LANL cluster for 14 days and 9 minutes, i. e. 336:09:00 hours, and nodes crashed according to the log file [CFDR 08]. In this time period, 42 resource crashes occurred. Figure 9.5 shows the downtimes of the compute nodes which conform to the repair times of the nodes. Some outliers have to be removed in order to show that usually the downtime is less than 3:36 hours and often less than 1.5 hours (see Figure 9.6). Downtimes which last only a few minutes are rare.

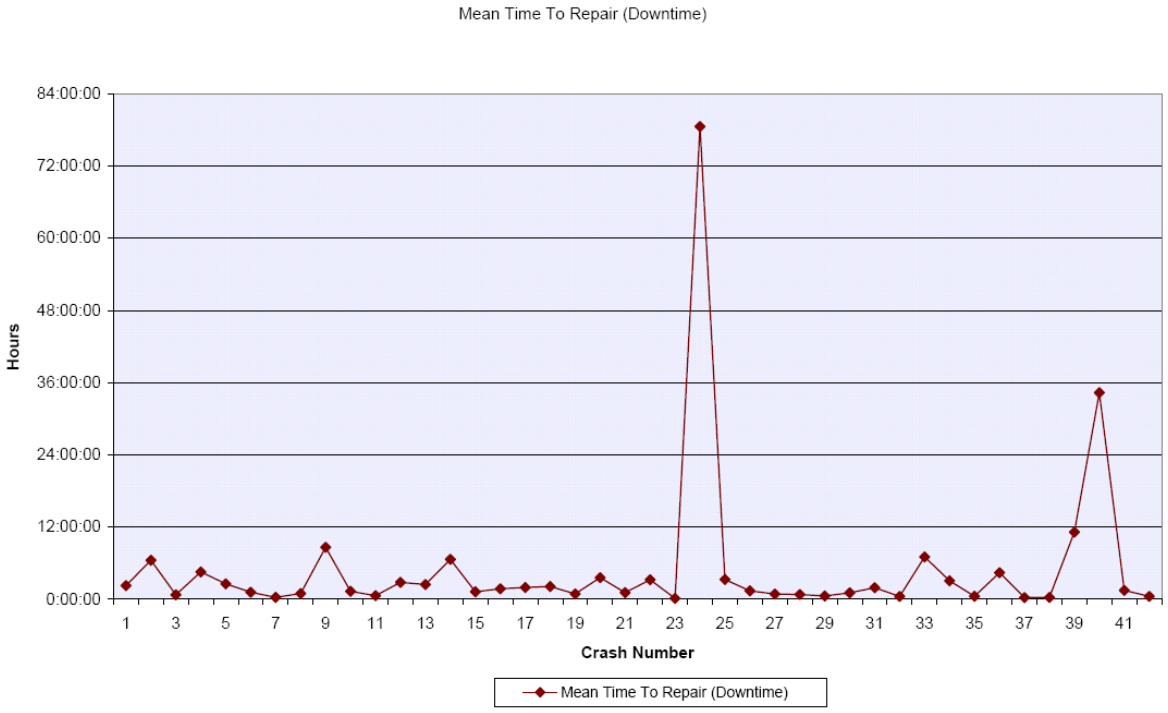


Figure 9.5: Down/Repair-Times as Logged

Revenue and penalty fee have been defined according to the definitions in Section 9.3.2. Two different categories have been defined for the standard PoF:

PoFS = 0.1 % jobs using 128 or more nodes

PoFS = 0.05 % jobs using less than 128 nodes

Such standard PoFs can be found based on long-term observations in a commercial Grid environment. The market price was set to 1.0€ for using one CPU for one hour. In order to achieve a good schedule, it was necessary to define a long execution windows which was five times longer than the duration, i. e. the difference between deadline and earliest start time equals to 5·duration. In order to not consider tremendous high penalty fees, the ratio of buffer

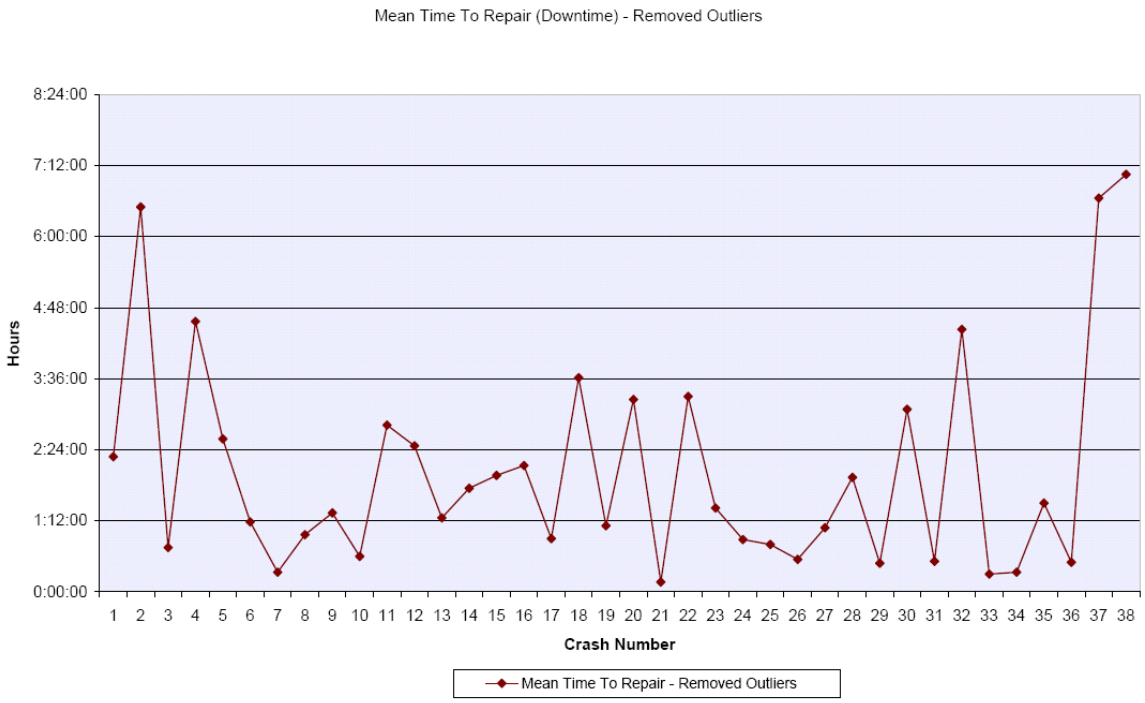


Figure 9.6: Removed Outlier from Down/Repair-Times

to duration was factorised by $f = 0.5$. These assumptions result in the following penalty definition:

$$\text{Penalty}(j) = \left(\frac{5 \cdot \text{duration}}{\text{duration}} \cdot 0.5\right) \cdot \text{duration} \cdot \text{nodes} \cdot 1.0\,€ \cdot \frac{(1 - \text{PoF})}{(1 - \text{PoFS} \in \{0.05, 0.1\})} \quad (9.4)$$

Figure 9.7 depicts the revenues and penalty fees of the CM-5 jobs defined according to the resource consumption and PoF.

The PoF calculations for the jobs are shown in Figure 9.8. Jobs in the LANL trace use in most cases either 32, 64, or 128 nodes. The maximum PoF for jobs using 128 or more nodes was requested to be 15%, and 0,07% otherwise. These hard limits can be identified in the distribution of PoFs calculated for the jobs. Some jobs have to reserve dedicated spare resources in order to achieve this upper bound. If jobs had to reserve dedicated spare resources, in most cases the FT-reservation consisted only one resource. As a consequence, the profit of the provider was not significantly reduced.

The Risk Management strategy to use spare nodes which are either dedicated or assigned to a pool, is an important means to prevent SLA violations. Figure 9.9 presents a job-centred view of how often such resources have been used in order to compensate for resource failures. Note that running jobs use first all the dedicated resources they have, then they try to find free resources on the cluster which are not assigned to the pool. If following both strategies have not found an/enough alternative resource(s), a resource of the spare pool is used. Thus,

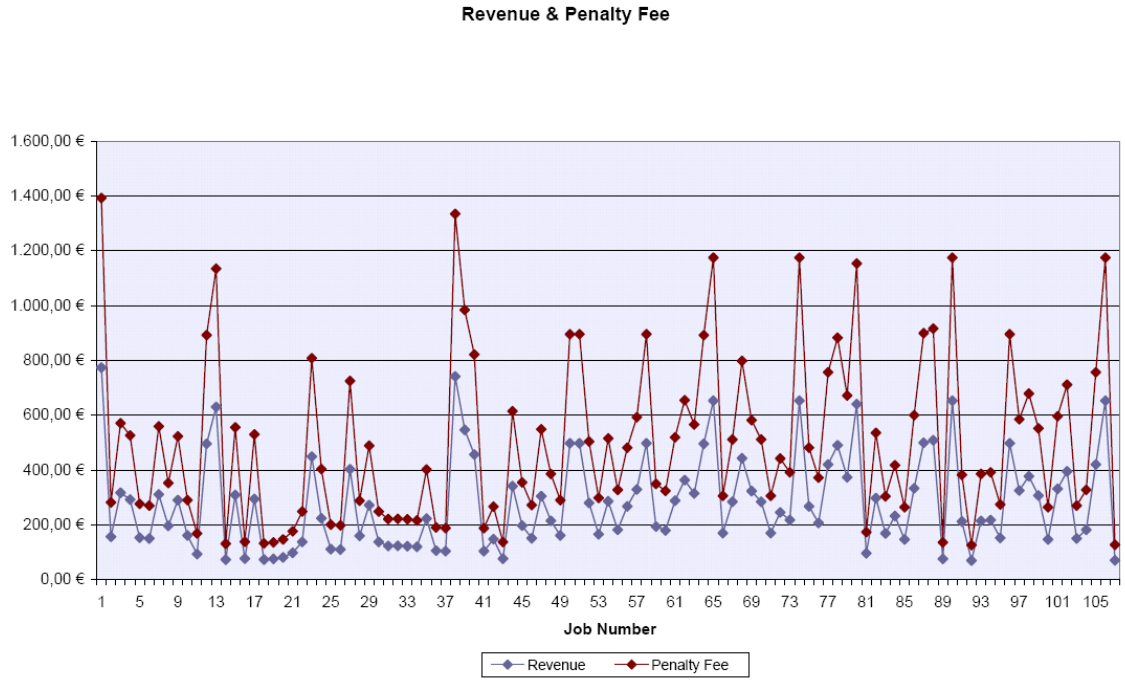


Figure 9.7: Revenue and Penalty Fees

these figures do not contain the usage of alternative free resources which are not assigned to the pool and available because of a lower system utilisation.

From the commercial perspective financial factors are crucial, Figure 9.10 depicts the revenue and penalty fees for those jobs used spare nodes. The sum of the penalty fee equals to the loss the provider would have if it would not use this Risk Management strategy.

In the case that not enough spare resources have been available to compensate for all resource outages, the provider generated a new schedule. Jobs have been prioritised according to their expected profit and loss as presented in Section 8.4.2. In order to ensure that the schedule does not completely differ from the previous version, the estimated priority value has been increased for jobs which have been scheduled to run at the time of evaluation. In the simulations the comparison value was factorised with 1.5. The jobs which could not be inserted into the schedule again, have been the in expectation least profitable ones. Figures 9.11 – 9.13 show the details about jobs have been affected by the resource outage and which jobs were not inserted in the schedule again. These jobs might be outsourced to other Grid providers. The success of a negotiation with another Grid provider is hard to estimate since this is influenced on market mechanisms and system utilisation. As a consequence, the results do not reflect the usage of other providers and a total loss depends on the negotiations for outsourcing. The cost for outsourcing depends on the PoF requested as well as the remaining execution time of the job, which might correspond to the complete job duration if the job has not started yet. If the provider does not ask for a different PoF it has accepted from its service consumer, than the revenue the provider pays for outsourcing is not higher than the revenue it receives. In particular, the revenue for outsourcing is lower if the job had been executed already and a

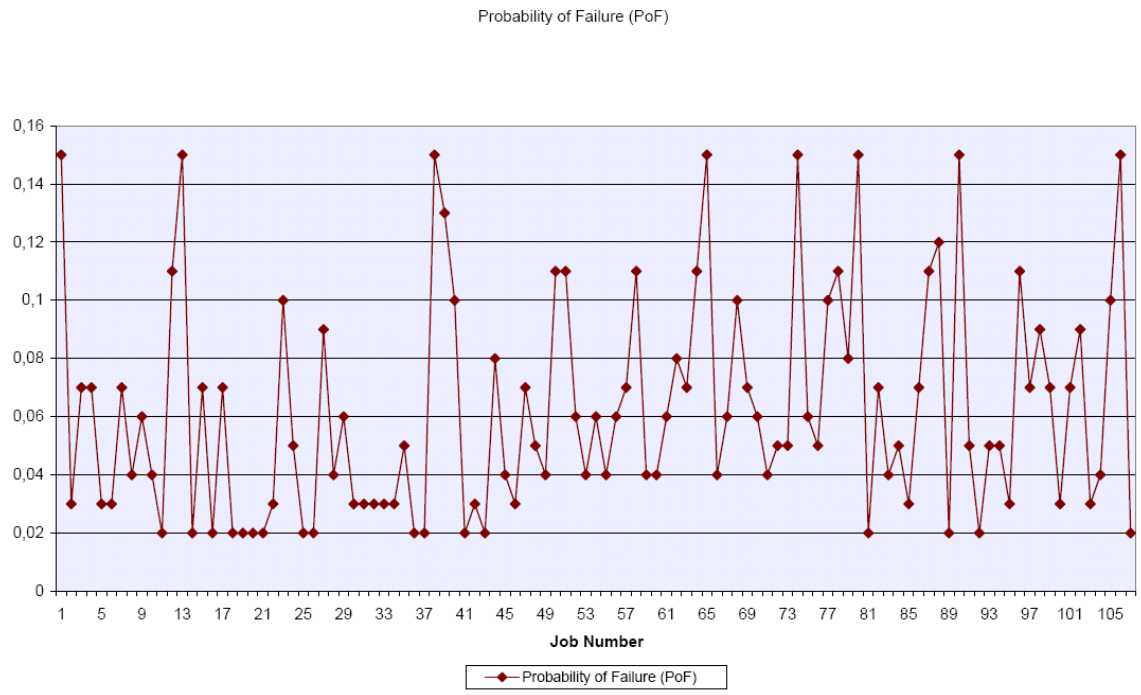


Figure 9.8: PoF Calculated

checkpoint is available. If the provider asks for the same penalty fee it had committed to, the complete risk is transferred to the other provider.

9.5 Other Scenarios and Parameters

The benefit of using Risk Management in the provider's resource management may be considered by varying the conditions of the basic scenario described in Section 9.3. This section shows some modifications which might be used for other evaluations.

The impact of different definitions of revenue and penalty fee can be compared by running the exact simulation twice and compare the resulting profit and loss. Note that for performing a simulation exactly twice it is necessary to submit the same jobs of the log trace in the same order to the RMS. Furthermore the resource outages have to occur on the same time, i. e. the same log trace has to be used.

An interesting aspect is also to vary the size of the spare pool in order to point out the difference in the PoFs which are estimated during the SLA negotiation. The usage of a spare pool is considered in the results of the basic scenario. Note that the loss which would result without the pool need not to be the same as the penalty fees of jobs used the pool (as depicted in Figure 9.10). Finally if compensating for all resource outages is not possible, a complete rescheduling is initiated and the in expectation least profitable jobs are migrated or the SLAs are violated.

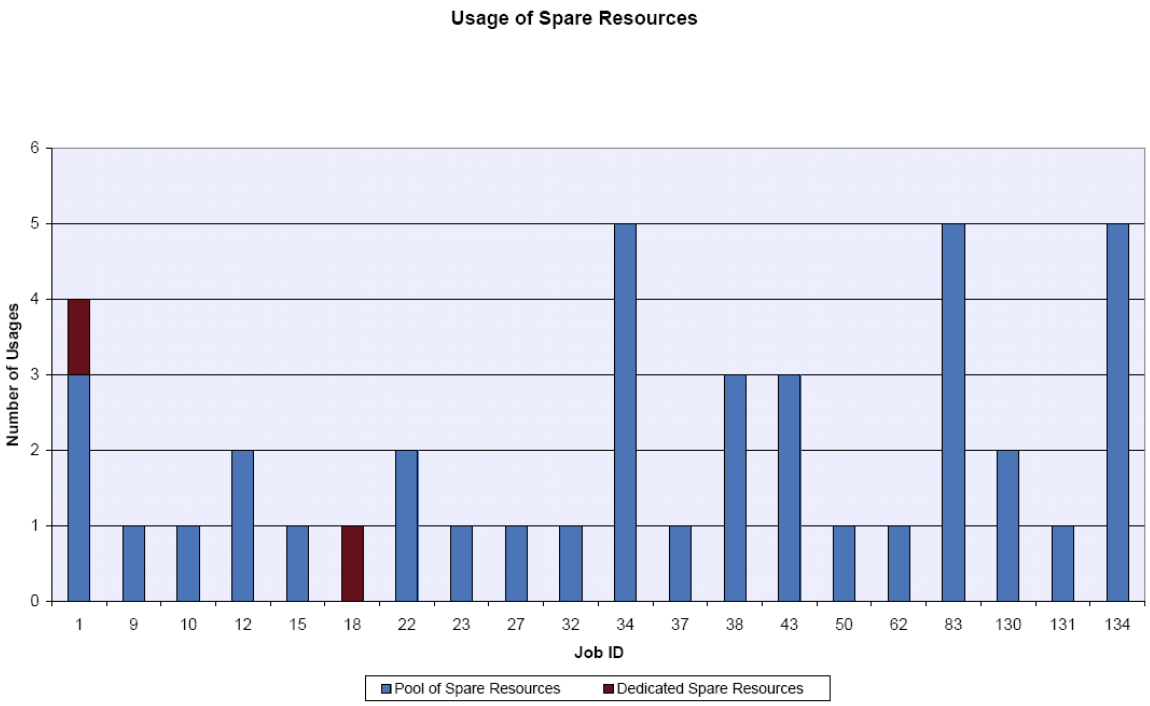


Figure 9.9: How many Jobs have used Spare Nodes?

The basic scenario initiates checkpointing from the beginning. In this context, different approaches can be tested. The cost of additional resources used to realise checkpointing should be calculated in order to evaluate its cost-effectiveness. This will point out how important is a default initiation of checkpointing.

To evaluate the benefits of migration, it is necessary to run simulations on various clusters in parallel. The providers have to be able to negotiate SLAs according to the risk aware WS-Agreement approach described in Section 9.1.2. Dependent on the workload of each provider, a negotiation will be successful. In this scope various scenarios can be tested when simulating a dynamic Grid market, since providers might define different PoFs and revenues when performing outsourcing.

9.6 Contextualise Results with Risk Management Definitions

The developed mechanisms and strategies are related to Risk Management. In order to point out that they correspond to definitions in Risk Management standards, this Section compares the requirements for a Risk Management process with the results. Section 9.6.1 uses a general perspective, whereas Section 9.6.2 compares approaches in the scope of IT-Risk Management with the developments.

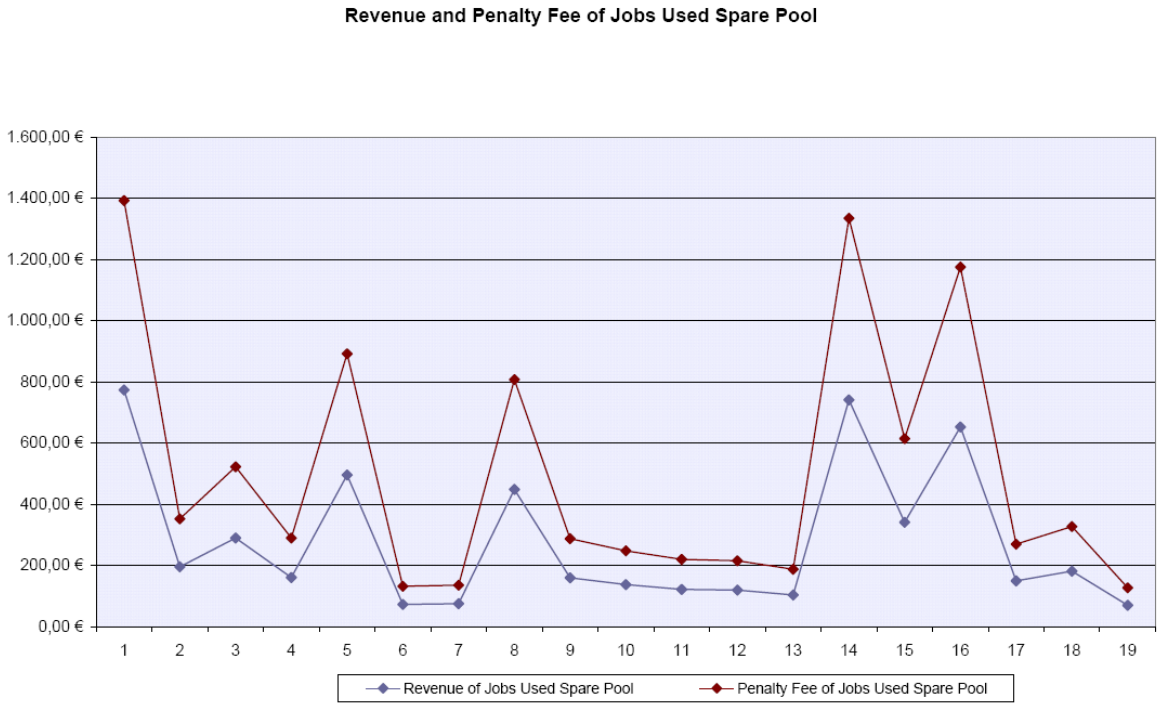


Figure 9.10: Revenue and Penalty Fee of Jobs Used Spare Nodes of the Pool

9.6.1 Risk Management in General

"Risk Management is a central part of any organisation's strategic management. ... It increases the probability of success, and reduces both the probability of failure and the uncertainty of achieving the organisation's overall objectives." [FERMA 03]

Risk Management supports to achieve the organisation's objectives by:

- Providing a framework for an organisation that enables future activity to take place in a consistent and controlled manner
- Improving decision making, planning, and prioritisation by comprehensive and structured understanding of business activity, volatility, and project opportunity
- Contributing of more efficient use/allocation of capital and resources within the organisation
- Optimising operational efficiency
- Protecting and enhancing assets and company reputation

The framework to enable an organisation – in this work acting as Grid provider – to perform their activities in a consistent and controlled manner is realised by introducing risk awareness into the provider's resource management. The decisions to be made in scope of Risk Management are clearly defined for the SLA negotiation, scheduling, and initiation of FT-mechanisms. Due to their integration as part of the automated SLA provisioning process,

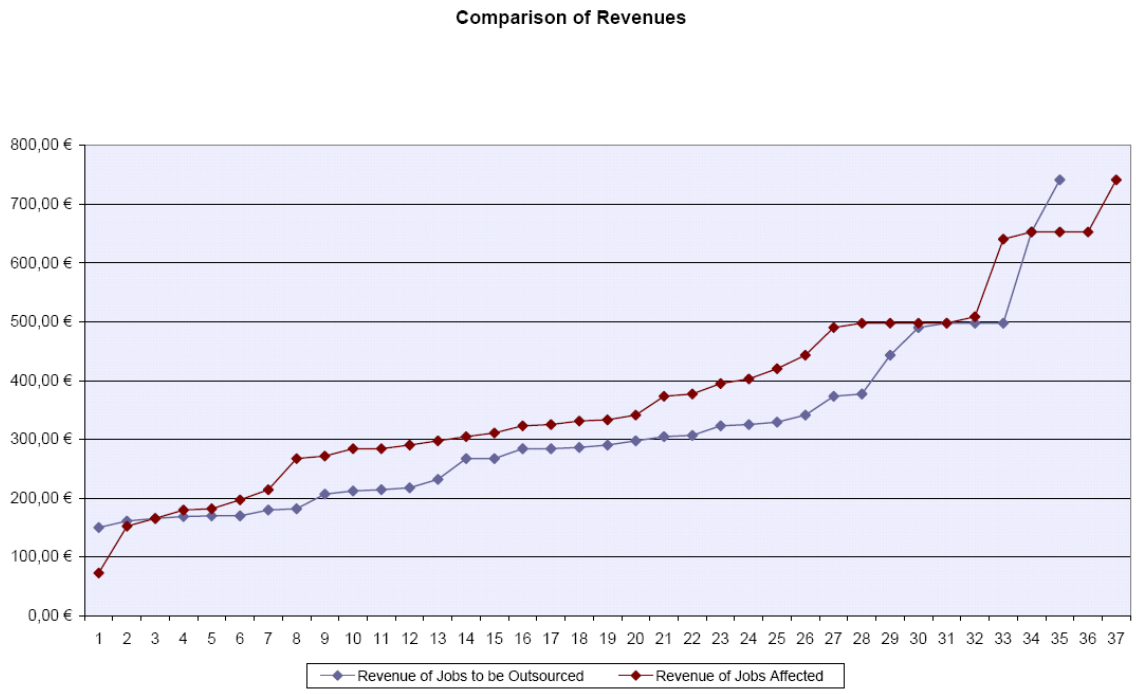


Figure 9.11: Revenues of Jobs Affected by a Resource Outage and Those to be Outsourced

these are controlled and consistent. The risk awareness integrated in the provider's resource management further improves the decision making in the SLA negotiation and in the post-negotiation phase. In particular the planning of resource utilisations and prioritisation of jobs is important in the post-negotiation phase if not sufficient resources are available. Chapter 7 and 8 present the developments of the Risk Management in these phases. Since in these decision processes the expected profit the provider will make is a key aspect, the integrated Risk Management contributes to a more efficient use of resources. In particular, the operation cost for the cluster, staff, etc. increases the efficiency of resource utilisation. The comparison performed during the SLA negotiation of the estimated PoF and the maximum PoF, which the customer is willing to accept, provide means to protect damages of reputation.

Risk Management should be executed systematically starting with a detailed identification and definition of each risks and its categorisation [OMah 05]. For this work this was performed in a risk identification which considered the threats for an SLA and the processes of an SLA violation. Results of the risk identification can be found in Chapter 5 (Section 5.4) as well as in Chapter 4 which is completed by [Mold 06].

The expected loss for each risk has to be estimated dependent on the definition of threats. Note that direct losses can be measured, however this is difficult for indirect losses, for example, caused by damage of the reputation. The loss considered in this work in scope of SLA provisioning is the associated penalty fee. However, several notes have been given to also consider the damage of reputation which should be taking into account by appropriate policies.

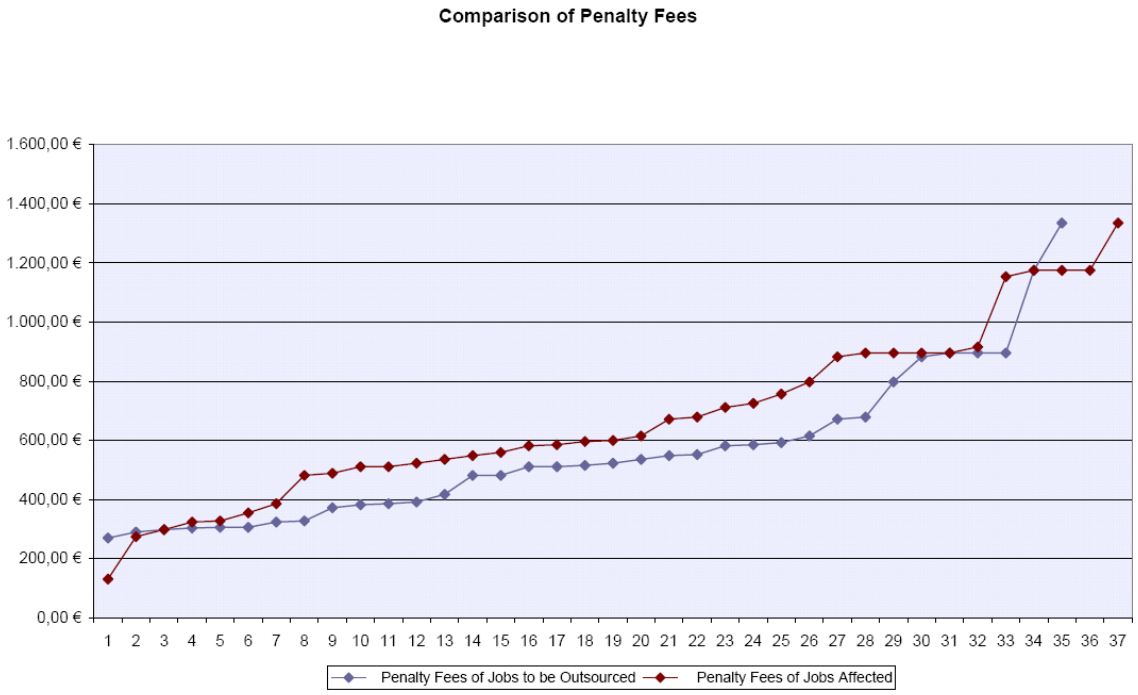


Figure 9.12: Penalty Fees of Jobs Affected by a Resource Outage and Those to be Outsourced

9.6.2 IT- Risk Management

Nowadays, nearly every companies involves an IT-infrastructure to execute their business workflow. In scope of IT-Risk Management, the following risks have been defined [Ober 05].

Operational risk is the classical IT risk and describes the danger of failures or non-availability of the IT caused by defects or security vulnerability

Business risk lacking flexibility to following market requests

Financial risk risk caused by unpredictably cost or by not used opportunities

Legal risk for example software piracy

This work focuses on operational risks since these are the most important IT risks. Furthermore the financial risk is considered since an SLA includes a reward and penalty fee and the resource provider has to decide whether to agree or reject an SLA request. Accepting an SLA might result in a loss in the case of an SLA violation, however, rejecting an SLA request implies to not use the opportunity to receive the reward for the service provisioning.

In the IT Risk Management not the technology itself should be assessed [Ober 05]. In the framework of this work this means that no reliability of performance estimations should be made about the Grid technologies used for the service provisioning, such as the Grid middleware, the RMS, etc. It is more important to assess the impact of the IT component, i. e. the compute nodes, for the productivity and conclude from this its importance of functionality. According to [Ober 05] risk influences four different business sections:

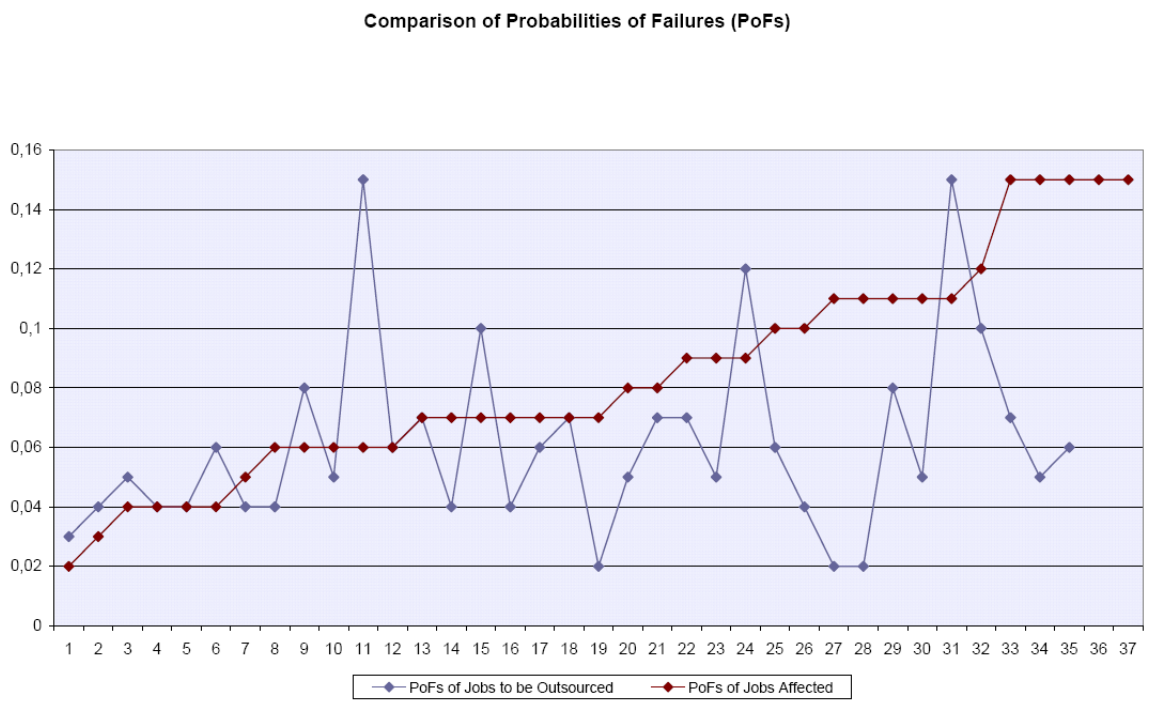


Figure 9.13: PoFs of Jobs Affected by a Resource Outage and Those to be Outsourced

- Interference of the service provisioning (performance)
- Damage of reputation (outside perspective)
- Financial effects (cost)
- Violation of laws/regulation results in penalty payments

In this work each of these four business sections are addressed. When compute outages occur, jobs cannot be executed as planned. Even if the Risk Management is able to compensate for these failures, the total performance is usually reduced if resources have failed. Jobs which have been planned to be completed earlier will be delayed. If this delay does not result in an SLA violation, no financial loss arises, however, the performance of the service provisioning is lower.

The damage of a Grid provider's reputation was kept in mind during this work. It should be considered in policies as well as in the estimation of the expected loss an SLA violation will result in. This work has defined the expected loss as the penalty since in general the damage of reputation cannot be determined well in figures for arbitrary organisations. Additionally, the damage of the reputation has to take into account the service owner and the requirements of the Grid job. Both aspects will influence the damage since stronger requirements or stricter customers might ask for a more reliable SLA provisioning.

The financial factors in the context of SLA provisioning are the cost for performing FT-mechanisms in order to compensate for resource failures. This cost is considered in the decision

whether to initiate FT-mechanisms or to accept the SLA violation.

The last business section listed is defined on two different levels. First, penalty payments have to be paid on a job basis if the associated SLAs are violated. Second, the high level view of acting according to laws and regulations is important. However, these are not part of the Grid Risk Management process which supports the resource management in scope of SLA provisioning.

Chapter 10

Related Work

The idea of integrating risk awareness and Risk Management in Grid processes has not been worked out before. As a consequence, related work, which addresses the support for SLA provisioning in the Grid by using Risk Management functionalities, does not exist. Hence, this chapter cannot present tightly related work and gives a more broadly overview of developments in scope of Risk Management and SLA provisioning is given.

As the central point of this work is introducing Risk Management in the Grid, the overview of related work starts in Section 10.1 with presenting currently applied IT-Risk Management in companies. The focus of the risk assessment is the stability of resources since their outages are the biggest threats for an SLA violation. Section 10.2 presents the two most important analysis in scope of computational Grids. In addition the results of a study about disk failures are summarised. To complete the overview of related work, contemporary solutions in scope of SLA management in the Grid are described in Section 10.3.

10.1 IT-Risk Management in Companies

The classical field for the identification and assessment of risks are financial services. Insurance companies developed and introduced in the 1960ies and 1970ies systematical mathematical methods in order to analyse and assess risks [OMah 05]. Nowadays, IT risks form a significant aspect in business processes since these companies increasingly dependent on IT. Due to the tight link of business processes and IT, a failure in the IT-infrastructure is more likely to impact the business than in the past. Since the IT is often a main technology used, the impact is particularly severe. Especially for companies with IT based business models, such as e-Sellers, failures in the IT infrastructure can easily become threatening its existence [Ober 05]. In former days, companies have equal IT risk with security. However, a new study of Symantec Corp. [Symantec 08] shows that companies are now aware that IT risk encompass security, availability, performance, and compliance elements. Furthermore the trend moves from performing Risk Management as a reactive or annual project to an continuous process. This is seen as of particular importance since changes in the IT are likely and accordingly a continuous Risk Management has to be performed.

IT-Risk Management processes in companies are acting on a higher level than the Risk Management developed in this work. The IT-Risk Management processes focus on the complete business workflows including service delivery, accounting, billing, law compliance etc. As a

consequence, IT-Risk Management is often considered from the perspective of organising and managing business processes. Since nowadays plain Grid resource providers do not exist, the business workflows and service delivery usually depend on the IT-infrastructure but are completed by additional services. To consider IT risks in the context of business workflows, the *Information Technology Infrastructure Library* (ITIL) framework [ITIL 07], part of the *IT Service Management* (ITSM) process, as well as the ISO/IEC 17799 security and various audit standards are applied. The high response of applying ITIL – more than 20 percent of billion dollar companies have completed at least one ITIL implementation – reflects the importance of IT-Risk Management for their business. Note that the implementation of standards and audit certificates are often a consequence in order to proof reliability.

In addition to security and compliance risk, the unavailability or underperformance of IT are threats for realising business workflows [Symantec 08]:

Availability Risk that information or applications will be made inaccessible by process, people or system failures, or natural disasters.

Performance Risk that underperforming systems, applications, staff or organisations will diminish business productivity value.

Since Grid service providers do not exist yet, availability risks and performance risks in companies have usually only internal effects: reduced revenue, added expense, or lost profit. The impact of unavailability and underperformance of the IT is high and expensive to cover [Symantec 08, Dyne 06]. Accordingly, implementing backup and recovery systems of data or compute centres forms a means to compensate for the unavailability of IT. However in most organisations, i. e. $> 70\%$, activating these systems takes more than four hours [1] which implies high losses for the company. Note that banks usually operate dedicated data centres and activating the backup only takes some seconds or minutes. However realising this fail-over is very expensive and often not acceptable for various companies.

Since IT-Risk Management becomes an urgent issue nowadays, companies such as Microsoft or SAP[2] offer service to support customers in evaluating risks and building IT-Risk Management plans. These are however very specific for the customer's business, infrastructure, organisation, and processes. In addition, internal Risk Management plans are confidential and consequently, no specific information is available how such IT-Risk Management plans are implemented. Some examples extracted from the description of the *Microsoft Operations Framework* (MOF) Risk Management Discipline [MOF 04] show that IT-Risk Management is however acting on a superior level than the Risk Management of this work:

Risk Acceptance Consider a data centre which needs to temporarily house servers in a basement room which is at risk of flooding. Finding an alternative to reduce the risk or transfer the risk would be too expensive. If additionally the room has not been flooded before, it might be justifiable to accept this risk, use the basement room, and monitor the situation.

Risk Transference Consider a company operating an e-Commerce web site. Instead of verifying credit cards on their own, they may outsource this tasks to another company. The risk, that the credit card is invalid, still exist but the outsourcing leads to that the

[1] http://www.zdnet.de/security/news/0,39029460,39154244,00.htm
[2] SAP SI Competence Center Risk Management & IT Security

partner company is responsible for this risk. If the partner is an expert in the field of credit verification, the risk might be reduced by outsourcing.
Note that other common examples for risk transference is effecting insurance or using external consultants with greater experience.

Risk Mitigation/Reduction Consider a company which uses a redundant network connections to the Internet in order to reduce the probability of losing access by eliminating the single point of failure. As shown in [Dyne 06] the productivity often significantly depends on the Internet connectivity and consequently such risk mitigation plans will be applied in various companies.

These examples show that current IT-Risk Management in companies are on a higher level than the results of this work. They consider complete business workflows which usually includes staff responsibilities. Hence, these are not executed automated and only some plans are implemented to increase fail-overs. Mechanisms applied generally in IT-Risk Management in scope of risk reduction are well comparable with this work. A prioritisation of tasks/service delivery will probably exist in internal IT-Risk Management in order to privilege urgent/profitable tasks. However, details of the concrete implementation of IT-Risk Management plans are not available since the process are company specific and also confidential. The developed decision processes addressing the estimation of PoFs and the initiation of FT-mechanisms may be integrated as a risk reduction strategy in superior IT-Risk Management plans.

10.2 Analysing Stabilities of Resources

This work is motivated by the fact that resources are unstable and fail during service provisioning which is a threat of fulfilling SLAs. Studies have been performed which analyse the resource availabilities in different Grids. These have been used to model and validate the PoF estimation process. Two important works have to be mentioned in this field. First of all, Section 10.2.1 presents an overview of monitoring data and analysis of the Grid'5000. An extensive collection of monitoring data is available from the *Los Alamos National Laboratory* (LANL), Section 10.2.2 details their observations. In addition to these statistics about resource availability, an analysis about hard disk failures is interesting in the scope of the general Risk Management process. Section 10.2.3 gives a summary of this study made by Google.

10.2.1 Grid'5000

Grid'5000 is an experimental Grid platform consisting of nine sites (Grid virtual organisations) geographically distributed in France. Each site comprises one or several clusters, for a total number of 15 clusters and over 2.500 processors. Each cluster is made of a set of bi-processors nodes. Figure 10.1 shows the structure of Grid'5000.

Iosup et al. [Iosu 07] analysed availability traces recorded by all batch schedulers handling Grid'5000 clusters, from mid-May 2005 to mid-November 2006. Altogether, this trace collected more than half a million of individual events that occurs on nodes. Each event in the trace represents a change in the status of nodes: either a node becomes available or unavailable.

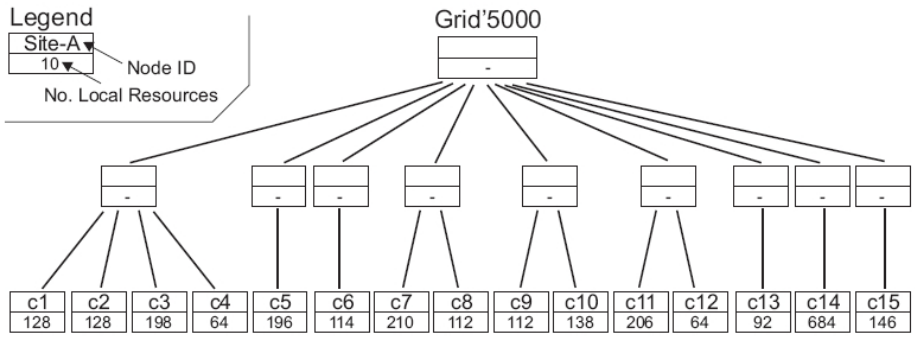


Figure 10.1: Structure of the Grid'5000 (Number of Processors per Cluster))

On average, resource availability in Grid'5000 at this level is 69% (±11.42), with a maximum of 92% and a minimum of 35%. The *Mean Time Between Failures* (MTBF) of the environment is of 744 ± 2631 seconds, i. e. around 12 minutes. At a cluster level, resource availability varies from 39% up to 98% across the 15 clusters. The average MTBF for all clusters is 18404 ± 13848 seconds, i. e. around 5 hours. As expected, this value is much higher than the MTBF at the Grid level. At a node level, the analysis showed that on average a node fails 228 times (for a trace that spans over 548 days). However, some nodes fail only once or even never according to the results. The duration of a failure is defined as the time elapsed between the occurrence of the failure, and the recovery of the resource affected by the failure. The duration of availability is defined in a similar way: the average availability duration of a node is 161315±113678 seconds (45 hours), whereas the average failure duration is of 51375±48267 seconds (14 hours). In both case, the standard deviation is quite high: it is almost equal to its associated value. The high standard deviations appear throughout the material; a closer analysis shows that the resource availability has to be treated on as a time-nonhomogeneous stochastic process, cf. Figure 10.2:

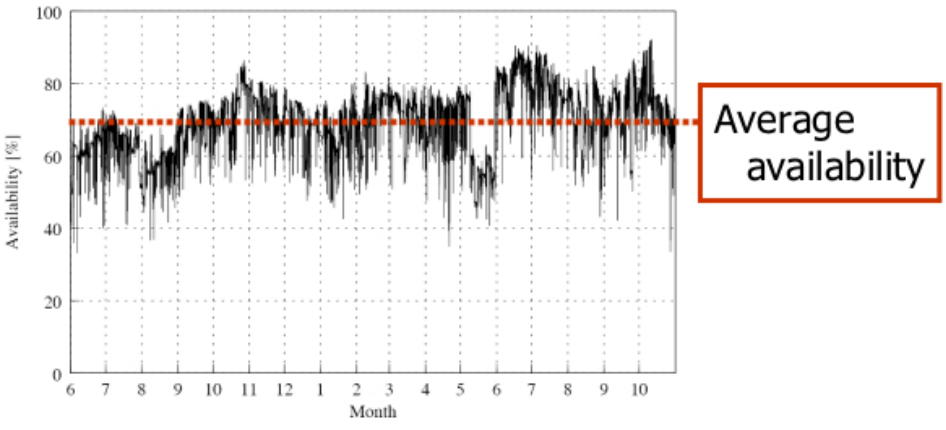


Figure 10.2: Resource Availability at Grid Level for Grid'5000

It is important to consider that for the failure duration, values may include night hours during which administrators of sites of a Grid are not available. Furthermore, some node failures

may be caused by a hardware detect and consequently a hardware exchange is necessary. For instance, since a processor or a memory slot has to be replaced, failure durations are higher than usually. The inter-arrival time of availabilities/failures, at the node level, is defined as the time between two consecutive availability/failure events on the same node. The material collected from Grid'5000 shows that the average inter-arrival time of availabilities or failures is 212848 ± 121122 seconds (59 hours).

In most cases of Grid failures the occurrence has been treated as independent events, i. e. nodes fail independently, which makes for easier modelling and less demanding data collection. Nevertheless, Iosup et al. investigated the notion of correlated failures, i. e. how a single failure (either a node or a set of nodes) can affect other nodes. Their analysis shows that on average the size of a correlated failure is 11.0 ± 21.0, with a maximum of 339. This latter value is little less than the size of the largest cluster, which is made of 342 nodes. To confirm this value, they analysed the number of sites involved in a correlated failure and found that correlated failures generally do not expand beyond a site; they also found that correlated failures represent less than 30% of the total number of failures events in the trace (around 300k). Their findings show that in further work on dynamic RA, the possibility of correlated failures has to be considered.

Iosup et al. continued by building models to describe the resource availability for a Grid; this was done by fitting statistical distributions to the availability data. They tried several distributions – Normal, Log-Normal, Exponential, Weibull, and Gamma – and fitted them to the data using the *Maximum Likelihood Estimation* (MLE) method. This was followed by goodness-of-fit tests to assess the quality of the fitting for each distribution, and to establish a best fit for each of the model parameters. They found that the best fits for the inter-arrival time between failures, the duration of a failure, and the number of nodes affected by a failure, are the Weibull, and Log-Normal. The results for inter-arrival time between consecutive failures was found to be alarming: the shape parameter of the Weibull distribution is (high) above 1, which indicates an increasing hazard rate function (the frequency with which a system or component fails, provided that it has survived so far). This indicates that the longer a computing node stays in the system, the higher the probability of the node's failure, preventing long jobs from finishing. This feature was considered in the risk assessment model (see Section 6.2) should be included in RA models.

For Risk Management purposes the evaluation of Grid/cluster performance becomes important. The performance depends on many factors, of which Iosup et al. identifies the system's architecture, the workload, and also the system's and the user's objectives (these factors are similar to the results shown by Schroeder and Gibson [Schr 06]). The objectives may be different for different actors in the Grid environment: resource providers may have as their objective to maximise the number of jobs completed for a specific user or to maximise the utilisation of the whole system and users may have as their objective to complete as many jobs as possible during a fixed time interval, or getting the jobs started with as little waiting time as possible. As Iosup et al. points out that collecting availability-aware performance statistics is necessary to implement Risk Management methods and strategies.

10.2.1.1 Difference to the Risk Assessment Model Developed

The difference of the model of Iosup et al. and the underlying risk assessment model in this work (described in Section 6.2) is that Iosup et al. do not use a probability function to describe the failure rates. Their functions which model the failure probabilities only depend on fixed input parameters describing the failure rate. An advantage of the use of the multinomial distribution in this context is its ability to represent any type of multimodal distributions, in contrast to the standard parametric families, such as the Geometric, Negative Binomial and Poisson distributions. For instance, if there are two major classes of computing tasks or maintenance events, such that one class is associated with relatively small numbers of required nodes, and the other with relatively large numbers, the system behaviour in this respect can be well represented by a multinomial distribution. On the other hand, standard parametric families of distributions would not enable an appropriate representation, unless some form of a mixture distribution were utilised. Such a choice would complicate the inference about the underlying parameters due to the fact that the number of mixture components would be unknown *a priori*.

10.2.2 Los Alamos National Laboratory (LANL)

Schroeder and Gibson [Schr 06] analyse failure data recently made publicly available by one of the largest high-performance computing sites. The data has been collected over the past 9 years at *Los Alamos National Laboratory* (LANL) and includes 23.000 failures recorded on more than 20 different systems, mostly large clusters of SMP and NUMA nodes. Their study includes root cause of failures, the mean time between failures, and the mean time to repair. They found that average failure rates differ wildly across systems, ranging from 20-1000 failures per year, and that time between failures is modelled well by a Weibull distribution with decreasing hazard rate. From one system to another, mean repair time varies from less than an hour to more than a day, and repair times are well modelled by a log-normal distribution.

The LANL site has hosted a diverse set of systems. Systems vary widely in size, with the number of nodes ranging from 1 to 1024 and the number of processors ranging from 4 to 6152. Systems also vary in their hardware architecture. There is a large number of NUMA and SMP based machines, and a total of eight different processor and memory models. The nodes in a system are not always identical. While all nodes in a system have the same hardware type, they might differ in the number of processors and *Network Interfaces* (NICs), the size of main memory, and the time they were in production use. This is probably typical for Grid computing environments, however the nodes of one single compute cluster are in most cases homogeneous in all these aspects.

At LANL a failure record contains the time when the failure started, the time when it was resolved, the system and node affected, the type of workload running on the node and the root cause. Root causes fall in one of the following five high-level categories: Human error; Environment, including power outages or A/C failures; Network failure; Software failure; and Hardware failure. In addition, more detailed information on the root cause is captured, such as the particular hardware component affected by a Hardware failure. With sufficient data on the root causes over sufficient periods of time it will be possible to build cause-effect models for failures and then build predictions on these models.

Schroeder and Gibson [Schr 06] characterise their empirical material with three import metrics: the mean, the median, and the squared coefficient of variation (C2). The squared coefficient of variation is defined as the squared standard deviation divided by the squared mean. They also use the empirical *Cumulative Distribution Function* (CDF) and how well it is fit by four probability distributions commonly used in reliability theory: the exponential, the Weibull, the Gamma, and the Log-Normal distribution. They use MLE to parameterise the distributions and evaluate the goodness of fit by visual inspection and the negative log-likelihood test. The approach they use is standard and similar to the methods used in this risk assessment model (see Section 6.2).

Fault-tolerance is frequently implemented through periodic checkpointing. When a node fails, the job(s) running on it is stopped and restarted on a different set of nodes, either starting from the most recent checkpoint or from scratch if no checkpoint exists. This approach is the same as in the underlying model of this work (see Section 6.1.4).

When failures occur, hardware was found to be the single largest cause, with the actual percentage ranging from 30% to more than 60%. Software is the second largest contributor, with percentages ranging from 5% to 24%. It is important to note that in most systems the root cause remained undetermined for 20-30% of the failures. Since in all systems the fraction of hardware failures is larger than the fraction of undetermined failures, and the fraction of software failures is close to that of undetermined failures, Schroeder and Gibson could still conclude that hardware and software are among the largest contributors to failures. However, they could not conclude that any of the other failure sources (human, environment, network) is insignificant.

Schroeder and Gibson [Schr 06] also found that the yearly failure rate varies widely across systems, ranging from only 17 failures per year for one system, to an average of 1159 failures per year for several other systems. In fact, variability in the failure rate is high even among systems of the same hardware type. The main reason for the vast differences in failure rate across systems is that they vary widely in size. The same characteristics can be found in the Grid'5000 data (see Section 10.2.1). Schroeder and Gibson [Schr 06] further found that there are some nodes which make up only 6% of all nodes but that they account for 20% of all failures. A possible explanation is that these nodes run different workloads than the other nodes in the system. They have made similar observations for other systems, where failure rates vary significantly depending on a node's workload. This observation is contrary to our assumption of Poisson failure rates, but requires more data and more systematic study.

Schroeder and Gibson next look at how failure rates vary across different time scales, from very large (system lifetime) to very short (daily and weekly). Knowing how failure rates vary as a function of time is important for generating realistic failure workloads and for optimising recovery mechanisms. By looking at the failure rate over the entire lifetime of a system they found that the failure rate actually grows over a period of nearly 20 months, before it eventually starts dropping. One possible explanation for this behaviour is that getting these systems into full production was a slow and painful process. By looking at the monthly failure rate they found that failure rates are high initially, and then drop significantly during the first months. The shape of this curve is intuitive in that the failure rate drops during the early age of a system, as initial hardware and software bugs are detected and fixed and administrators gain experience in running the system. Next they look at how failure rates vary over smaller time scales. It is well known that usage patterns of systems vary with the time of the day

and the day of the week. The question is whether there are similar patterns for failure rates: they observe a strong correlation as during peak hours of the day the failure rate is two times higher than at its lowest during the night. Similarly the failure rate during weekdays is nearly two times as high as during the weekend. They interpret this as a correlation between a system's failure rate and its workload, since in general usage patterns (not specifically LANL) workload intensity and the variety of workloads is lower during the night and on the weekend. Another possible explanation would be that failure rates during the night and weekends are not lower, but that the detection of those failures is delayed until the beginning of the next (week-) day.

Schroeder and Gibson also study the sequence of failure events as a stochastic process and study the distribution of its inter-arrival times, i. e. the time between failures. They take two different views of the failure process: (i) the view as seen by an individual node, i. e. they study the time between failures that affect only this particular node; (ii) and the view as seen by the whole system, i. e. they study the time between subsequent failures that affect any node in the system. They found that from 2000-2005 the distribution between failures for individual nodes is well modelled by a Weibull or gamma distribution. Both distributions create an equally good visual fit and the same negative log-likelihood. For the system wide view of the failures the basic trend for 2000-2005 is similar to the per node view during the same time. The Weibull and gamma distribution provide the best fit, while the log-normal and exponential fits are significantly worse.

10.2.3 Failures of Hard Disks

Pinheiro et al. [Pinh 07] carried out a study of data collected from a large number of disk drives which are deployed in several types of systems across all of Google's services. The total population was more than 100.000 disk drives, which are a combination of serial and parallel ATA consumer-grade hard disk drives, ranging in speed from 5400 to 7200 rpm, and in size from 80 to 400 GB; the units were put in production in or after 2001. The conclusions are interesting: (i) there is no consistent pattern of higher failure rates for higher temperature drives or for those drives at higher utilisation levels (which is a partial contradiction of the results of Schroeder and Gibson); (ii) after the first scan error drives are 39 times more likely to fail within 60 days than drives with no such errors; (iii) first error in reallocations, off-line reallocations and probational counts are also strongly correlated to higher failure probabilities. These observations point to data to look for when monitoring data is collected from Grid and cluster operations.

10.3 SLAs in Grids

This section present details to the current usage and consideration of SLAs in Grids. An overview is given in Section 10.3.1. Afterwards in Section 10.3.2 and Section 10.3.3 the focus is set on solutions for Grid brokers and providers.

10.3.1 Overview

The *Service Negotiation and Acquisition Protocol* (SNAP) was developed to manage remote SLAs [Czaj 02] negotiated with different resource providers. It addresses the problem of reserving resources before job submission for applications, which require resources on demand, in order to achieve the overall objective of the Grid to provide a transparent means for fulfilling end-users' application requirements. This is implemented by using three different types of SLAs:

Task Service Level Agreements (T-SLA) By a T-SLA the service steps and resource requirements for the execution of a tasks are negotiated. A T-SLA is created by submitting a job description to the provider.

Resource Service Level Agreements (R-SLA) A R-SLA is used to negotiate the usage or consumption of a resource. It must not specify for which purpose the resource should be used. For example, a R-SLA is an advance reservation.

Binding Service Level Agreements (B-SLA) B-SLAs are used to combine T-SLAs and R-SLAs, i. e. negotiating about the application of a resource for a task.

This proposed SLA model is independent of the service to be negotiated and consequently has many fields of application. The usage of this protocol for a broker solution has been realised in [Haji 05] by using a three-phase commit protocol. Sahai et al. stated in their work [Saha 03] that all quality aspects important for a commercial utilisation cannot be defined by using these notions. Since T-SLA, R-SLA, and B-SLA focus on the definition of task submission techniques and negotiations of time constraints, Sahai has described in [Saha 01] even complex SLAs by using the *Web Service Description Language* (WSDL) [Chri 01].

Resulting from the essential need for SLAs, the Grid community has identified the requirement for a standardisation of SLA description and negotiation. Within the Open Grid Forum (OGF) [OGF 08], this work has been driven by the *Grid Resource Allocation Agreement Protocol* (GRAAP) working group, resulting in two standard proposals for SLA description and the process of negotiation WS-Agreement [Andr 07] and WS-Agreement Negotiation [Andr 06]. Contract negotiations within distributed systems have been the subject of research regarding *Business-to-Business* (B2B) service guarantees. Approaches for mapping of natural language contracts onto models suitable for contract automation were proposed but similar mechanisms have neither been applied in a Grid environment nor as an SLA.

One of the first architectures supporting SLAs in the commercial Grid context was defined from Hewlett-Packard Laboratories [Saha 03] by considering the HP Utility Data Center as a typical commercial Grid deployment environment. The architecture relies on a network of proxies which have committed to an SLA and belong to different administrative domains. The SLA management in this conceptual architecture is defined as a OGSA meta service. In order to realise an SLA management, the interaction with several services of the Grid infrastructure is necessary as depicted in Figure 10.3, such as the registration and discovery service for finding appropriate resources conforming to the QoS requirements defined in the SLA.

To implement all functionalities required for an SLA management on this Grid layer, a set of protocols can be used. The Grid community has agreed to utilise established standards for this. For example, data transfer is usually realised in contemporary systems by Grid FTP; the

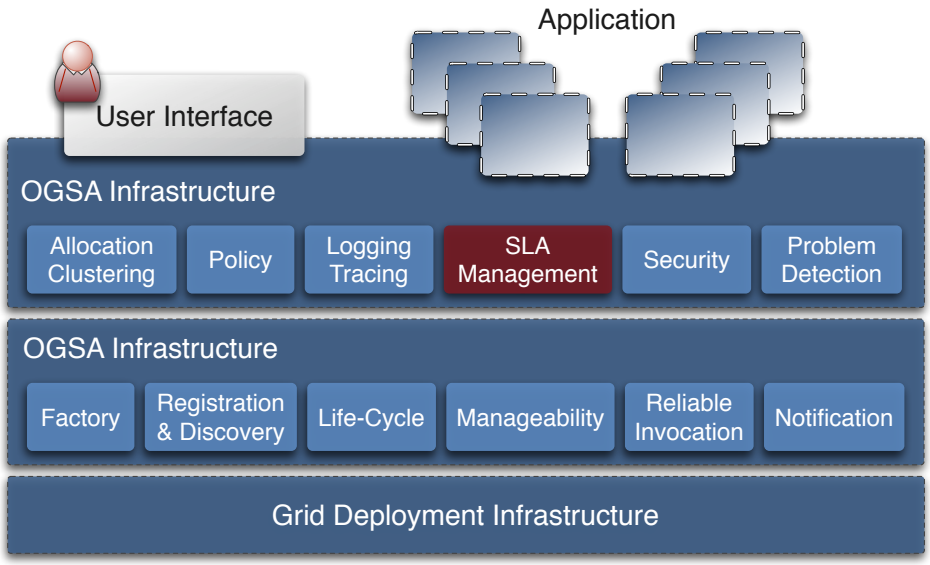


Figure 10.3: Conceptual Architecture Addressing SLA Management in Commercial Grids [Saha 03]

registration and discovery of services is performed by *Grid Information Service* (GIS), *Globus Security Infrastructure* (GSI) [Welc 03a] is used in scope of authorisation and authentication by using credentials. Protocols of the OGF's GRAAP group – namely WS-Agreement or WS-Agreement Negotiation – could be used for resource allocation and management.

The number of projects funded by the European Commission and focusing on the integration of SLAs in Grid services is a further indicator of the importance of this topic. Addressing SLAs in such projects is ranging from architecture/technology developments as in NextGrid [NextGrid 08] or HPC4U [HPC4U 08], to the usage and negotiation of SLAs as in Akogrimo [Akogrimo 08], BEinGRID [BEinGRID 08], or OntoGrid [OntoGrid 08], up to considering SLAs from the economic perspective as in GridEcon [GridEcon 08].

10.3.2 Brokers

Access to Grid resources should be realised in a transparent manner. The resource utilisation of one provider is realised by a RMS. However, specific applications, such as large distributed simulations [Brun 98] or on-line experiments [Lasz 99], require that resources distributed in the Grid are useable *simultaneously*. Scheduling decisions concerning resources over multiple administrative domains is defined as resource brokering [Scho 02]. Hence, Grid brokers are core components in scenarios to establish a transparent Grid utilisation if allocated resources are operated by different Grid providers. They analyse the specification, discover suitable resources, and map jobs to selected resources. Since these resources are operated by autonomous providers, a Grid broker must be capable of fulfilling the matching process without direct control of underlying resources. The support of advance reservations as well as the negotiation of SLAs with providers are important means for brokers (see Section 10.3.4.1).

Nimrod/G was developed in order to realise an automated modeling and execution of parameter studies in the computational Grid by using distributed resources. Hence, it is a Grid resource broker providing resource discovery, trade for resources, scheduling, steering and data management [Buyy 00a]. The trading of resources depends on supply and demand and reacts to dynamic changes. In order to realise the automated execution of parameter studies, it dispatches jobs to remote Grid nodes, starting and managing their job execution, and gather results back to the home node. In the GRACE (Grid Architecture for Computational Economy) framework it considers economic aspects in order to dynamically trade for end-users a better service and support the consideration of budget and deadline constraints [Buyy 00b]. The underlying economic model of GRACE includes in addition to the global scheduler (broker) also a bid-manager, directory server, and bid-server. A close interaction with Grid middleware and Grid fabric solutions is required to fulfill their tasks. Nimrod/G is primarily used with Globus Toolkit but can also interact with Legion, Condor, or NetSolve.

Nimrod/G enables users to trade off QoS parameters, deadlines and computational cost. The economic resource scheduling focuses on resource cost defined by the provider, price the user is willing to pay, and a deadline. In addition to these economic aspects, the scheduling mechanism has a multitude of parameters which have to be considered. The list of parameters address in particular QoS issues like the resource architecture and configuration, the speed or memory size of the compute nodes, access speed, network bandwidth, etc. Furthermore, the consideration of the number of free or available nodes, the priority of the user, the scheduling queue length and type as well as the reliability of the resources are crucial in order to met the deadline. All these parameters require a detailed knowledge of the resource states and hardware in the Grid fabric. This work addresses some of these parameters by using a planning-based scheduler which notifies the Grid broker about the planned execution time. Furthermore, the reliability of the resources are defined even more precisely by offering a PoF of the SLA fulfillment. Consequently, if combining Nimrod/G with the results of this work, the Grid broker must not estimate such values without a precise knowledge.

Another common used broker interacting with Grid middleware or RMS solutions such as Globus, Legion, and NetSolve is, for example, the *Application Level Scheduling* (AppLes) [Berm 96] which is based on an agent implementation. To allocate appropriate resources, it considers dynamic as well as static application and system information. By using the Network Weather Service AppLes is notified about changes in the performances of resources dynamically. Such monitoring information is essential in order to prevent SLA violations: if resources provide a lower performance as required, jobs can be migrated to other resources in order to prevent an SLA violation. However, note that such features are only possible, if providers grant access to the required monitoring information.

A challenge for brokers is that the Grid is dynamic and providers are not willing to disclose their infrastructure, resource utilisation, policies, etc. Consequently, published information from the providers may be limited, incomplete, or stale. If various providers offer the same performance, current resource brokers cannot estimate which provider will be the most reliable. As an example, the cheapest provider will not necessarily be the best one with respect to trustworthiness and risk of overloads. Consequently, getting clear knowledge of the PoF per provider and SLA would facilitate and support its decision making.

Complying with the idea of publishing PoF information within an SLA, is the new task of

the broker in which it estimates the reliability of such an estimation. [Gour 08] describes the approach used in scope of the AssessGrid project to mark providers as reliable, moderate or unreliable. Based on this classification a provider's PoF information is adjusted in order to notify end-users about accurate values.

10.3.3 Providers

A key component of resource providers is their RMS. SLA support has to be integrated in it in order to allocate resources which conform to SLA requirements and to prevent SLA violations. According to this tight relationship between RMS and SLAs, this section does not focus only on SLA provisioning (see Section 10.3.4.1), but also gives an overview of contemporary RMSs (see Section 10.3.4).

10.3.4 Resource Management Systems

At the Grid fabric layer, RMSs are managing the resources of one cluster and realise that users can access the Grid infrastructure. There is a broad landscape of RMSs, both commercial and non-commercial, having their individual orientation and audience. Furthermore, established RMS for clusters have been modified for their application in Grids. Maui [Maui 08] is a popular plug-in scheduler for RMSs such as Portable Batch System (OpenPBS) [OpenPBS 08] or Sun Grid Engine [SGE 08]. Its popularity is caused by its capabilities in the areas of advance reservations, extensive resource availability query support, external job migration, resource charging, and QoS support. At present, the *Load Sharing Facility* (LSF) [LSF 08] is the market leader for commercial RMSs. It comprises load sharing and batch queuing software that manages, monitors, and analyses the resources and workloads on a network of heterogeneous computers. Condor/G [Condor 08] is a widespread non-commercial RMS, allowing users to take advantage of both dedicated and non-dedicated computers. It allocates resources based on parameters that enhance system utilisation and throughput. Condor/G can manage resources for its jobs but provides no support for co-allocation, site autonomy, or the heterogeneous nature of Grid systems. All existing developments do not use PoF information in the negotiations, scheduling, and resource management.

The EC-funded project HPC4U (*Highly Predictable Clusters for Internet Grids, IST-511531*) [HPC4U 08] is implementing an SLA-aware Grid fabric. The key results of this project are the integration of the FT-mechanisms checkpointing and migration in a planning based RMS. Their checkpointing solution is in particular useable without recompiling the software which enables the usage of checkpointing also for commercial applications. The developments of this work will complement the achievements of HPC4U since risk awareness is essential in scope of the initiation of fault-tolerance mechanisms if not sufficient resources are available.

10.3.4.1 Provisioning of SLA/QoS

Afzal et al. [Afza 08] state that Grid computing infrastructures build on a cost-effective computing paradigm that virtualises heterogeneous system resources to meet some dynamic

needs of critical business and scientific applications. They also find that Grid computing environments are inherently dynamic and unpredictable environments sharing services among many different users. In order to use the services a scheduling of the resources which should meet two (sometimes conflicting) objectives is required: (i) to make the most efficient use of Grid resources (high utilisation) while (ii) providing the best possible performance to the Grid applications. In meeting both objectives it is necessary to satisfy a set of associated performance and QoS constraints. In commercial Grid settings there is a further need to work out schedules which will minimise the execution costs for running the tasks but still meeting the QoS constraints. When handling resource failures in this work, the cost and benefit of a schedule are in the main focus. This objective is considered within the Risk Management process since the expected profit and the loss are compared from different SLA bound jobs in order to select the most profitable ones.

In order to be able to offer QoS provisioning, a RMS has to support advance reservations [Al A 04]. Such advance reservations in the RMSs are important from the perspective of the Grid middleware since mapping a parallel job execution on several Grid sites, time constraints are crucial in order to ensure that all sub-jobs are executed in parallel. In scope of workflow jobs, time guarantees have to be negotiated in order to ensure that the dependencies of several sub-jobs are fulfilled. This work is build on a planning-based system supporting advance reservations. In particular, the mechanism of making an advance job reservation is crucial for the PoF information. Hence, to integrate risk awareness into different RMS, they should support this concept.

Hovestadt has developed an SLA scheduling mechanism by using a planning based RMS [Hove 06b]. The fulfilment of deadlines is evaluated during the SLA negotiation by considering the fact stated in [Yeo 05] that it is necessary to balance competing service requests, while ensuring that agreed levels of service performance are achieved. His work considers the usage of FT-mechanisms in order to prevent SLA violation in the case of resource failures. The developed strategies show that either enough own resources are available to perform an FT-mechanism successfully or outsourcing is possible. If neither of those assumptions are fulfilled, the SLA of the job would be violated which was affected by the resource outage. However, this is not the most profitable solution and generating a new schedule in which the job would be violated or outsourced which results in the lowest loss is important. Furthermore, it is essential to consider PoFs in the scheduling during and after the SLA negotiation since these are decisive for accepting or rejecting an SLA and for the expected profit.

Libra [Sher 04] implements also a deadline-aware scheduling and requests therefore a runtime estimation of the user as common in planning-based systems. The cost for a job execution is defined based on the execution time as well as the buffer the end-user allows according to the deadline, i. e. the ratio of execution time, earliest start time, and deadline. A big difference to the approach realised in Libra and in the used planning-based system OpenCCS is that in Libra more than one job can use the same resource. In OpenCCS a compute node is dedicated available for one job execution. In a commercial Grid environment, this implementation will be essential since customers are not willing to share resources simultaneously because of security issues.

LibraSLA [Yeo 05] considers in the resource allocation the deadline and the penalty. In particular, Yeo and Buyya differ between hard and soft deadlines, i. e. in contrast to a hard deadline, a soft deadline describes that the user can accommodate a delay. Furthermore they

define a penalty rate for compensating the loss of the end-user by a decreasing function. This decreasing function depends on time and reduces the reward after the deadline. The underlying model of the LibraSLA is very similar to this work, for instance LibraSLA considers in addition to the deadline, the number of resources as well as the runtime of the job as main QoS criteria. However, LibraSLA also follows the concept of processor sharing as the original Libra [Sher 04] solution does. This work determines the job priority according to the utility of the job by considering the runtime and deadline. According to the utility, the return of a job is calculated and the job with the highest return has the highest priority to be executed. The work lacks of considering PoFs which are essential to take into account when estimating the profit the provider will earn if it execute the job. In particular, the necessity of considering the expectation value by taking into account PoFs is shown by the analysis of the resource availability presented in Section 10.2. Furthermore, the benefits of using outsourcing mechanisms have not been addressed.

Chapter 11

Conclusion

The intention when developing the Grid was to provide location and time independent, transparent, and pervasive access to compute and data resources for arbitrary Grid users. Although the technologies required to support such access exist, Grids are predominately used in the scientific/academic environment or as Enterprise Grids, i. e. internal within one organisation. Using a commercial Grid infrastructure would be cheaper for organisations to run compute-intensive applications than to operate their own resources. Consequently, there is a need to support commercial Grid usage on a per job basis. However commercial end-users typically operate their own resources at present. One of the most important reasons why they are unwilling to make use of a commercial Grid environment is that they have to give away the control on their applications. In order to offer end-users reliable service provisioning and reduce their concerns in using the Grid, obligations and *Quality of Service* (QoS) aspects for the job execution can be defined in *Service Level Agreements* (SLAs). SLAs are powerful instruments to describe all aspects of a contract between service provider and service consumer: ranging from service description up to guarantees and contractual penalties. However, the definition of SLAs is only a concept and not an established tool. This may be the main reason for the lack of commercial Grid uptake. The lack of acceptance of a system based on SLAs is a consequence of two conflicting objectives: first, end-users will be only persuaded to use the Grid instead of operating their own resources if providers agree to meet strict QoS requirements. However, Grid providers are unwilling to commit to such obligations since resource outages and job failures are common. The attitude of resource providers is understandable when considering failure rates of current Grids: in DAS-2 more than 10% of all jobs fail [Iosu 06], in TeraGrid the failure rate is $10 - 45\%$ [Khal 06], and in Grid3 27% jobs fail even with 5-10 retries [Dumi 05].

The work presented here addresses this problem for the Grid provider by integrating Risk Management into its processes. Integrating risk awareness during SLA negotiation enables Grid providers to trade off the risk of committing to an SLA with the profit they might gain if they accept it. It is crucial to integrate risk awareness in the reservation process of resource management in order to estimate the *Probability of Failure* (PoF) of an SLA during negotiation. A specific reservation process which focuses on the reservation process might be used. Alternatively, risk awareness can be combined with arbitrary (established) scheduling strategies by not modifying the reservation process itself and rather add risk considerations on a higher layer. If the estimated PoF for an SLA is too high, the provider may apply risk reduction strategies to lower the risk to an acceptable level. When planning risk reduction, it is necessary to consider the profit margin the provider requires since the usage of *Fault-Tolerance*

(FT)-mechanisms always results in additional cost.

In the SLA post-negotiation phase the objective of a Grid provider is to fulfill its obligations in order to meet the SLA. When a resource instability has been identified through monitoring, precautionary FT-mechanisms may lead to the prevention of an SLA violation since acting before a resource outage occurs saves valuable time. For instance a job j' using an unstable resource might be migrated to another resource *before* the job outage. Since applying an FT-mechanism has an impact on other jobs, it is necessary to evaluate the benefits of initiating it. Since FT-mechanisms may as be initiated jobs j, which will be affected by the FT-mechanisms for job j', recursive checks should be performed. If so many resource outages occur that spare resources cannot compensate for these, some SLAs have to be violated. In order to maximise the expected profit/minimise the expected loss, the provider has to prioritise jobs according to their expected profit and loss. If jobs cannot be executed on the provider's own resources, it may have the opportunity to outsource the job. The success of outsourcing depends, however, on the willingness of other providers to commit to the outsourcing of an SLA. Due to the uncertainty as to whether SLA negotiation with other providers will be successful, it is important to outsource those jobs whose SLA violation results in the lowest expected harm. Furthermore outsourcing the 'least important' jobs will be the more profitable since the outsourcing cost will be lower if requesting lower requirements.

Speaking in terms of Risk Management, in the SLA negotiation risk acceptance corresponds to committing to the SLA, whereas in the post-negotiation phase it means accepting an SLA violation. A risk avoidance strategy is only applicable during the SLA negotiation since it implies rejecting the SLA offer. Risk reduction/mitigation is achieved through the initiation of FT-mechanisms both during the negotiation and in the post-negotiation phase. Risk transference can be achieved by outsourcing a job and only makes sense in the post-negotiation phase since during SLA negotiation the SLA offer can still be rejected.

The evaluation results have shown the benefits of using Risk Management in the SLA provisioning. By using spare resources which are either dedicated to one job or assigned to a pool, resource failures can be handled without an impact on other jobs. The *dedicated* spare resources are used in order to lower the PoF for an SLA and to achieve a value which is lower than the maximum acceptable PoF which is specified from service consumers. Note that providers could lie or publish inaccurate PoFs, however, at the Grid broker layer reputation centres or services validating these values will exist. The pool of spare resources can be used by *any* SLA bound job and, if no resource failures need to be compensated, best-effort jobs might run on them. It is important to consider the number of spare resources assigned to the pool in order to ensure that not too many resources are in the pool, yet still enough to compensate for most resource failures. Monitoring data collected during SLA provisioning will help to identify the expected number of resources required to compensate for resource failures. This number can be used to determine the size of the spare pool.

To apply Risk Management in the Grid, it is necessary to realise an automated process which differs from the implementation of Risk Management in other fields of application. The mechanisms and decision processes presented in this work form the basis for specific solutions for Grid providers. By using them, Risk Management becomes an essential part of resource management and is frequently initiated in contrast to contemporary IT-Risk Management. This tight coupling of Risk Management and resource management supports providers significantly

in their SLA provisioning and enables them to estimate risks of committing to an SLA. Consequently, even if they are aware of the unreliability of Grid resources, they might be willing to accept strict requirements since the Risk Management addresses the issue of SLA violations.

List of Figures

1.1 Coarse-Grained Grid Architecture – Overview of Grid Layers and Resource Management . . . 6
1.2 By and By Job Allocation in Queuing-Based Systems . . . 6

2.1 Job Slot Bounded by Earliest Start Time and Latest Completion Time . . . 7
2.2 Problem of Customer Perspective – Will the SLA be Fulfilled? . . . 8
2.3 Problem of Provider Perspective – Will the SLA be violated? . . . 9

3.1 The Risk Management Process defined in FERMA Standard . . . 17
3.2 The Risk Management Process defined in AS/NZS Standard . . . 18
3.3 Relationship of Expected Loss and Probability of an Event . . . 23
3.4 Core Elements of the Open Grid Service Architecture [Fost 03] . . . 32
3.5 Structure of the WS-Agreement . . . 35
3.6 WS-Agreement Negotiation - State Machine [Andr 06] . . . 41
3.7 Abstract Structure of a Resource Management System [Krau 02] . . . 42

4.1 Overview of the KAOS Metamodel used for the Requirement Analysis . . . 47
4.2 High Level Objectives for a Risk Aware Grid . . . 50
4.3 Provider's Goal Model during Negotiation: Resource Selection . . . 53
4.4 Provider's Goal Model after Agreement: SLA Fulfilled at run-time . . . 54

5.1 FERMA Risk Management Process and Grid Risk Management Process Derived from FERMA. . . . 60
5.2 Targeted Grid Risk Management Process focusing on Using Risk as Decisive Factor . . . 67
5.3 Risk Identification – Origins and Problems for a Resource Failure . . . 70
5.4 Risk Identification – Origins and Problems for a Job Failure . . . 71

6.1 Assignments of Clusters and RMSs . . . 81
6.2 Job Slot Bounded by Earliest Start-time and Latest Finish-time . . . 82
6.3 Workflow of Processing an SLA Request . . . 84
6.4 Extended Job Duration caused by Checkpointing Initialisation . . . 86
6.5 2-Phases of SLA Provisioning . . . 96

7.1 How to Benefit from Risk Management During SLA Negotiation ? . . . 97
7.2 Increase Trust of Grid Actors by Using PoFs . . . 98
7.3 Risk Management Addressing Step during SLA Negotiation . . . 99
7.4 Reservation Process Focusing on PoF – Overview . . . 101
7.5 Resource Pre-Selection During Risk-Focusing Reservation . . . 103

7.6 Risk Aware Reservation Process . . . 104
7.7 Buffer for Performing FT-mechanisms equals Deadline - t_p . . . 105
7.8 Consider Time Differences when Counting Resultant Fragments . . . 111
7.9 Reduced Schedule Quality because of Small Fragments . . . 114
7.10 Example of Schedules of two Suitable Resources . . . 114
7.11 Risk Aware Reservation Process Coupled with Arbitrary Scheduling Strategies . 117
7.12 Highlighted Steps from Targeted Risk Management Process in Reservation Workflow . . . 119
7.13 Need to Extend Time to Handle Node Failures for Parallel Jobs . . . 121
7.14 Checkpointing Initialisation Results in Extension of Job Duration . . . 122
7.15 Risk Management Strategies in the Context of SLA Negotiation . . . 126
7.16 Are initial reservations useable during Renegotiation? . . . 128
7.17 Phases of SLA Provisioning - Focus Negotiation: Addressed Risk Management 131

8.1 How to Integrate Risk Management In Post-Negotiation . . . 134
8.2 Example Schedule for Recursive Evaluation of FT-Mechanism . . . 151
8.3 Example Schedule after Initiation of Migration for A and B . . . 153
8.4 Phases of SLA Provisioning - Focus Post-Negotiation: Addressed Risk Management . . . 163

9.1 Outcomes of the AssessGrid Project . . . 166
9.2 Implemented Workflow of SLA Negotiation . . . 169
9.3 Phases of SLA Provisioning - Results of Risk Management Integration . . . 171
9.4 Distributions of Jobs in the LANL CM-5 Log [CM 5 Tra 96] . . . 174
9.5 Resource Failures: Down/Repair-Times as Logged . . . 177
9.6 Resource Failures: Removed Outlier from Down/Repair-Times . . . 178
9.7 Revenue and Penalty Fees for Simulation . . . 179
9.8 PoF of Jobs Run in the Simulation . . . 180
9.9 Overview of how many Jobs have used Spare Nodes . . . 181
9.10 Benefit of Using Spare Nodes – Comparison of Jobs' Revenue and Penalty Fee . 182
9.11 Revenues of Jobs Affected by a Resource Outage and Those to be Outsourced . 183
9.12 Penalty Fees of Jobs Affected by a Resource Outage and Those to be Outsourced 184
9.13 PoFs of Jobs Affected by a Resource Outage and Those to be Outsourced . . . 185

10.1 Structure of the Grid'5000 (Number of Processors per Cluster) . . . 190
10.2 Resource Availability at Grid Level for Grid'5000 . . . 190
10.3 Conceptual Architecture Addressing SLA Management in Commercial Grids [Saha 03] . . . 196

List of Tables

3.1 Comparison of Queuing and Planning Based Systems [Hove 03] 43

8.1 Key Data of Exemplarily Evaluating Migration Effects 145
8.2 Evaluation Results of Exemplary Estimating Impact of a Migration 147
8.3 Evaluation Results of Migration With Interchanged Penalties 147
8.4 Recursive Evaluation of FT-Mechanisms – Example Schedule 151

9.1 PoF Estimations and Sampling Observations 172
9.2 Probabilities of Number of Node Crashes Affecting an Example Job 173

Chapter 12

Bibliography

[Afza 08] A. Afzal, A. S. McGough, and J. Darlington. "Capacity planning and scheduling in Grid computing environments". *Future Gener. Comput. Syst.*, Vol. 24, No. 5, pp. 404–414, 2008.

[Akogrimo 08] "Access Knowlegde through the Grid in a mobile World (Akogrimo) – Project Homepage". http://www.mobilegrids.org/, 2008.

[Al A 04] R. J. Al-Ali, K. Amin, G. von Laszewski, O. F. Rana, D. W. Walker, M. Hategan, and N. J. Zaluzec. "Analysis and Provision of QoS for Distributed Grid Applications.". *Journal of Grid Computing*, Vol. 2, No. 2, pp. 163–182, 2004.

[Alex 01] C. Alexander. *Market Models*. John Wiley & Sons, New Jersey, first Ed., 2001.

[Allc 00] W. Allcock, I. T. Foster, S. Tuecke, A. Chervenak, and C. Kesselman. "Protocols and Services for Distributed Data-Intensive Science". In: *Proceedings of the VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research*, pp. 161–163, 2000.

[Ande 81] T. Anderson and P. Lee. *Fault-Tolerance, Principles, and Practice*. Prentice Hall International, 1981.

[Andr 06] A. Andrieux, K. Czajkowsk, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. "Web Services Agreement Negotiation Specification (WS-AgreementNegotiation)– Version 1.0". Tech. Rep., November 2006. GRAAP WG - Open Grid Forum (OGF).

[Andr 07] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. "Web Services Agreement Specification (WS-Agreement)". http://www.ogf.org/documents/GFD.107.pdf, Mar 2007. GRAAP WG - Open Grid Forum (OGF).

[Anjo 06] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva. "Job Submission Description Language (JSDL) Specification – Version 1.0". November 2006. JSDL-WG – Open Grid Forum (OGF).

[ASNZS 99] "AS/NZS 4360: 1999 – Australian Standard Risk Management". 1999. Standards Australia, Standards New Zealand (AS/NZS).

[AssessGr 08] "Advanced Risk Assessment and Management for Trustable Grids (AssessGrid), EU-funded project IST-031772". http://www.assessgrid.eu, 2008.

[Aziz 07] B. Aziz, G. Silaghi, F. Martinelli, G. Dallons, and A. Arenas. "Reasoning about Trust and Security Properties in Dynamic Virtual Organisations: State of the Art". Tech. Rep., GridTrust, Apr 2007. Deliverable D2.1.

[Baga 07] E. Bagarinao, Y. Tanaka, and T. Nakai. "Building Grid-Based Applications for the Management and Analysis of Neuroimaging Data Sets for the Medical Grid". Vol. 25, No. 5, November 2007.

[Bart 05] W. Barth. *Nagios*. Open Source Press, München, 2005.

[Batt 07] D. Battré, O. Kao, and K. Voss. "Implementing WS-Agreement in a Globus Toolkit 4.0 Environment". *Usage of Service Level Agreements in Grids Workshop in conjunction with The 8th IEEE International Conference on Grid Computing (Grid 2007)*, 2007.

[Batt 08a] D. Battré, M. Hovestadt, O. Kao, A. Keller, and K. Voss. "Increasing Fault-tolerance by Introducing Virtual Execution Environments.". Tech. Rep. TR-RI 08291, University of Paderborn, Paderborn, Germany, 2008.

[Batt 08b] D. Battré, M. Hovestadt, O. Kao, A. Keller, and K. Voss. "Virtual Execution Environments for ensuring SLA-compliant Job Migration in Grids". *SCC 2008: International Conference on Services Computing, Honolulu, Hawaii, USA*, Jul 2008.

[BEinGRID 08] "BEinGRID – Project Homepage". http://www.beingrid.eu/, 2008.

[Benn 96] J. Bennett, G. Bohoris, E. Aspinwall, and R. Hall. "Risk analysis techniques and their application to software development". *European Journal of Operational Research*, pp. 467–475, 1996.

[Berm 96] F. D. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao. "Application-level scheduling on distributed heterogeneous networks". In: *Supercomputing '96: Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM)*, p. 39, IEEE Computer Society, Washington, DC, USA, 1996.

[Biet 06] M. Biette, F. Tétard, B. Fally, C. Ponsard, S. Mouton, J. Padgett, I. Gourlay, K. Djemame, K. Voß, and J. Stümke. "AssessGrid Deliverable D5.3 Preliminary Expoitation Plan". Tech. Rep., September 2006.

[Birk 07] G. Birkenheuer, P. Majlender, H. Nitsche, K. Voss, and E. Weber. "Gather and Prepare Monitoring Data for Estimating Resource Stability". *Proceedings of the Cracow Grid Workshop*, 2007.

[Blac 72] F. Black and M. S. Scholes. "The Valuation of Option Contracts and a Test of Market Efficiency". *Journal of Finance*, Vol. 27, No. 2, pp. 399–417, May 1972. available at http://ideas.repec.org/a/bla/jfinan/v27y1972i2p399-417.html.

[Boeh 89] B. W. Boehm, Ed. *Software risk management*. IEEE Press, Piscataway, NJ, USA, first Ed., 1989.

[Box 90] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[Brun 98] S. Brunett, D. Davis, T. Gottschalk, P. Messina, and C. Kesselman. "Implementing Distributed Synthetic Forces Simulations in Metacomputing Environments". In: *HCW '98: Proceedings of the Seventh Heterogeneous Computing Workshop*, p. 29, IEEE Computer Society, Washington, DC, USA, 1998.

[Buyy 00a] R. Buyya, D. Abramson, and J. Giddy. "Nimrod/G: An Architecture of a Resource Management and Scheduling System in a Global Computational Grid". *The Computing Research Repository (CoRR)*, Vol. cs.DC/0009021, 2000. informal publication.

[Buyy 00b] R. Buyya, J. Giddy, and D. Abramson. "An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications". In: *Proceedings of the 2 nd International Workshop on Active Middleware Services (AMS 2000)*, Kluwer Academic Press, Pittsburgh, USA, Aug 2000.

[Carm 02] M. Carman, F. Zini, L. Serafini, and K. Stockinger. "Towards an Economy-Based Optimisation of File Access and Replication on a Data Grid". In: *CCGrid 2002: 2nd IEEE International Symposium on Cluster Computing and the Grid, 22-24 May 2002, Berlin, Germany*, pp. 340–345, 2002.

[Carnegie 05] "Risk Management". Carnegie Mellon University - Software Engineering Institute, Carnegie Mellon – Office of the Under Secretary of Defense, 2005.

[Carr 93] M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker. "Taxonomy-Based Risk Identification". Tech. Rep. CMU/SEI-93-TR-6, Pittsburgh, Pennsylvania 15213, 1993.

[Catl 92] C. Catlett. "In Search of Gigabit Applications". In: *Communications Magazine*, pp. 42 – 51, Apr 1992.

[CFDR 08] "The Computer Failure Data Repository (CFDR) – USENIX – Homepage". http://cfdr.usenix.org/, 2008.

[Chri 01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. "Web services description language (WSDL) 1.1". Tech. Rep. ANL/MCS-P1000-1002, 2001.

[CM 5 Tra 96] "The Los Alamos National Lab (LANL) CM-5 log – Homepage". http://www.cs.huji.ac.il/labs/parallel/workload/l_lanl_cm5/index.html, 1996.

[Condor 08] "Condor-G Project". http://www.cs.wisc.edu/condor/condorg/, 2008.

[Copo 06] T. I. R. Coporation. "Grid Computing: A vertical market perspective 2005-2010". Press Release, 2006.

[COSO 04] "Enterprise Risk Management - Integrated Framework". September 2004. Committee for Sponsoring Organizations of the Treadway Commission (COSO).

[Cull 99] D. Culler, J. Singh, and A. Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann Publishers, San Fransisco, USA, 1999.

[Czaj 02] K. Czajkowski, I. T. Foster, C. Kesselman, V. Sander, and S. Tuecke. "SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems". In: *JSSPP '02: Revised Papers from the 8th International Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 153–183, Springer-Verlag, London, UK, 2002.

[Di M 07a] G. Di Modica, V. Regalbuto, O. Tomarchio, and L. Vita. "Dynamic renegotiations of SLA in service composition scenarios". *euromicro*, Vol. 0, pp. 359–366, 2007.

[Di M 07b] G. Di Modica, V. Regalbuto, O. Tomarchio, and L. Vita. "Enabling renegotiations of SLA by extending the WS-Agreement specification". In: *IEEE International Conference on Services Computing, 2007. SCC 2007*, pp. 248–251, 2007.

[Djem 06] K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, O. Kao, and K. Voss. "Introducing Risk Management into the Grid". In: *E-Science '06: Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*, p. 28, IEEE Computer Society, Washington, DC, USA, 2006.

[Dumi 05] C. Dumitrescu, I. Raicu, and I. T. Foster. "Experiences in Running Workloads over Grid3". In: *GCC – Grid and Cooperative Computing - GCC 2005, 4th International Conference, November 30 - December 3, 2005, Proceedings*, pp. 274–286, Springer, Beijing, China, 2005.

[Dyne 06] S. Dynes, E. Andrijcic, and M. E. Johnson. "Costs to U.S. Economy of Information Infrastructure Failures". Jun 2006.

[EGEE 08] "Enabling Grids for E-SciencE (EGEE) Project". http://www.eu-egee.org/, 2008.

[Elli 06] G. Elliott, C. Granger, and A. Timmermann, Eds. *Handbook of Economic Forecasting*. Vol. 1, Elsevier, 2006.

[Evan 00] M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. Wiley-Interscience, Jun 2000.

[FERMA 03] "A Risk Management Standard". 2003. Federation of European Risk Management Associations (FERMA).

[Fost 00a] I. Foster, A. Roy, and V. Sander. "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation". In: *8th International Workshop on Quality of Service*, 2000.

[Fost 00b] I. T. Foster and C. Kesselman. "Computational Grids". In: *VECPAR*, pp. 3–37, 2000.

[Fost 01] I. T. Foster, C. Kesselman, and S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". *International Journal of Supercomputer Applications*, Vol. 15(3), 2001.

[Fost 03] I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[Fost 05a] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. von Reich. "Open Grid Services Architecture (OGSA)– Version 1.0". Jun 2005. GFD-I.030 – OGSA WG – Open Grid Forum.

[Fost 05b] I. T. Foster. "A Globus Toolkit Primer. Or, Everything You Wanted to Know about Globus, but Were Afraid To Ask". Tech. Rep., Argonne National Laboratory, 2005.

[Fost 98] I. T. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastucture*. Morgan Kaufmann Publishers, San Fransisco, USA, first Ed., 1998.

[Gelm 02] A. Gelman. "Prior Distribution". In: *Encyclopedia of Environmetrics*, pp. 1634–1637, Chichester, 2002.

[Gelm 03] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, July 2003.

[Glat 06] T. Glatard, J. Montagnat, and X. Pennec. "An Experimental Comparison of Grid5000 Clusters and the EGEE Grid". In: *Workshop on Experimental Grid testbeds for the assessment of large-scale distributed applications and tools (EXPGRID'06)*, Paris, France, Jun 2006.

[Glob 97] "Globus: A Metacomputing Infrastructure Toolkit". *International Journal of Supercomputer Applications*, Vol. 11(2), pp. 115–128, 1997.

[Globus 08] "Globus Alliance: Globus Toolkit". http://www.globus.org, 2008.

[Gour 08] I. Gourlay, K. Djemame, and J. Padgett. "Risk and Reliability in Grid Resource Brokering". In: *Proceedings of IEEE International Conference on Digital Ecosystems and Technologies 2008 (DEST 2008)*, Phitsanulok, Thailand, Feb 2008.

[Grah 06] S. Graham and J. Treadwell. "Web Services Resource Properties 1.2 (WS-ResourceProperties)". http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf, April 2006. Document-Identifier: wsrf-ws_resource_properties-1.2-spec-os.

[Grid5000 08] "Grid'5000 – 5000 processors distributed in 9 sites France wide, for research in Grid Computing, eScience and Cyber-infrastructures". https://www.grid5000.fr/, 2008.

[GridEcon 08] "Grid Economics and Business Models (GridEcon) – Project Homepage". http://www.gridecon.eu/, 2008.

[GridTrus 08] "Trust and Security for Next Generation Grids (GridTrust), EC-funded project IST-033817 – Project Homepage". http://www.gridtrust.eu, 2008.

[GWA 08] "The Grid Workloads Archive – Homepage". http://gwa.ewi.tudelft.nl/pmwiki/, 2008.

[Haji 05] M. H. Haji, I. Gourlay, K. Djemame, and P. M. Dew. "A SNAP-Based Community Resource Broker Using a Three-Phase Commit Protocol: A Performance Study". *The Computer Journal*, Vol. 48, No. 3, pp. 333–346, 2005.

[Halb 74] H. Halberstam and H. E. Richert. *Sieve Methods*. Academic Press, London, UK, 1974.

[Hass 07] P. Hasselmeyer, B. Koller, I. Kotsiopoulos, D. Kuo, and M. Parkin. "Negotiating SLAs with Dynamic Pricing Policies". *Service Oriented Computing: a look at the Inside (SOC@Inside'07), Vienna, Austria*, pp. 28–32, Sep 2007.

[Hein 05] F. Heine, M. Hovestadt, O. Kao, and A. Keller. "Provision of Fault Tolerance with Grid-enabled and SLA-aware Resource Management Systems". In: *Parallel Computing: Current & Future Issues of High-End Computing, Proceedings of the International Conference ParCo 2005*, pp. 113–120, G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (Editors), John von Neumann Institute for Computing, Jülich, NIC Series, 2005.

[Hoel 71] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Statistical Theory*. Houghton Mifflin Company, Jun 1971.

[Hoff 02] D. G. Hoffman. *Managing Operational Risk: 20 Firmwide Best Practice Strategies*. Wiley, Feb 2002.

[Hove 03] M. Hovestadt, O. Kao, A. Keller, and A. Streit. "Scheduling in HPC Resource Management Systems: Queueing vs. Planning". In: *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP) at GGF8*, pp. 1–20, LNCS 2862, 2003.

[Hove 06a] M. Hovestadt. "Operation of an SLA-aware Grid Fabric.". *Journal of Computer Science*, Vol. 2, No. 6, pp. 550 – 557, 2006.

[Hove 06b] M. Hovestadt. *Service Level Agreement aware Resource Management*. PhD thesis, University of Paderborn, Fuerstenallee 11, 33102 Paderborn, Oct 2006.

[Hove 06c] M. Hovestadt, O. Kao, and K. Voss. "The First Step of Introducing Risk Management for Prepossessing SLAs". In: *SCC '06: Proceedings of the IEEE International Conference on Services Computing*, pp. 36–43, IEEE Computer Society, Washington, DC, USA, 2006.

[HPC4U 08] "Highly Predictable Cluster for Internet-Grids (HPC4U), EU-funded project IST-511531 – Project Homepage". http://www.hpc4u.org, 2008.

[IBM 06] "IBM Tivoli Workload Scheduler LoadLeveler". http://www.ibm.com/systems/clusters/software/loadleveler.html, 2006.

[Iosu 06] A. Iosup and D. Epema. "GRENCHMARK: A Framework for Analyzing, Testing, and Comparing Grids". *ccgrid*, Vol. 00, pp. 313–320, 2006.

[Iosu 07] A. Iosup, M. Jan, O. O. Sonmez, and D. H. J. Epema. "On the dynamic resource availability in grids". In: *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (GRID 2007)*, pp. 26–33, Austin, Texas, USA, Sep 2007.

[ISO 02] "Risk Management – Vocabulary – Guidelines for Use in Standards - ISO:IEC guide 73:2002". June 2002. International Organization for Standardization (ISO).

[ITIL 07] "IT Infrastructure Library (ITIL) Service Management Practices V3 Qualifaction Scheme". available at: http://www.itil-officialsite.com/nmsruntime/saveasdialog.asp?lID=168&sID=86, Nov 2007.

[Iver 84] G. R. Iversen. *Bayesian Statistical Inference (Quantitative Applications in the Social Sciences)*. Sage Publications, Inc, Nov 1984.

[JIS Q 01] "JIS Q 2001:2001 - Guidelines for development and implementation of risk management system". 2001. Japanese Standards Association.

[Kans 97] K. K"ans"al"a. "Integrating Risk Assessment with Cost Estimation". *IEEE Softw.*, Vol. 14, No. 3, pp. 61–67, 1997.

[Khal 06] O. Khalili, J. He, C. Olschanowsky, A. Snavely, and H. Casanova. "Measuring the Performance and Reliability of Production Computational Grids". In: *GRID – 7th IEEE/ACM International Conference on Grid Computing (GRID 2006), Proceedings*, pp. 293–300, IEEE, Barcelona, Spain, Sep 2006.

[Klei 69] L. Kleinrock. Jul 1969. UCLA Press Release.

[Koll 99] G. R. Koller. *Risk Assessment and Decision Making in Buisness and Industry – A Practical Guide*. CRC Press, Boca Raton, FLorida, USA, 1999.

[Krau 02] K. Krauter, R. Buyya, and M. Maheswaran. "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing". *Software — Practice and Experience*, Vol. 32, No. 2, pp. 135–164, 2002.

[Lams 91] A. van Lamsweerde, A. Dardenne, B. Delcourt, and F. Dubisy. "The KAOS project: Knowledge acquisition in automated specification of software". pp. 59–62, Mar 1991.

[Lasz 99] G. von Laszewski, J. A. Insley, I. T. Foster, J. Bresnahan, C. Kesselman, M.-H. Su, M. Thiébaux, M. L. Rivers, S. Wang, B. Tieman, and I. McNulty. "Real-time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Source.". In: *PPSC*, 1999.

[Lerc 06] N. Lerch, H. Nitsche, K. Voss, and M. Hovestadt. "First Steps of a Monitoring Framework to Empower Risk Assessment on Grids". *Proceddings of the Cracow Grid Workshop*, pp. 216–223, 2006.

[Leve 86] N. G. Leveson. "Software safety: why, what, and how". *ACM Comput. Surv.*, Vol. 18, No. 2, pp. 125–163, 1986.

[Lifk 95] D. A. Lifka. "The ANL/IBM SP Scheduling System". In: D. G. Feitelson and L. Rudolph, Ed., *Proc. of 1st Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 295–303, Springer Verlag, 1995.

[Liu 06] L. Liu and S. Meder. "Web Services Base Faults 1.2 (WS-BaseFaults)". http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf, April 2006. Document-Identifier: wsrf-ws_base_faults-1.2-spec-os.

[LSF 08] "LSF Homepage". http://www.platform.com/Products/platform-lsf-family, 2008.

[Majl 06] P. Majlender, C. Carlsson, F. Tétard, M. Heikkilä, I. Gourlay, K. Voss, G. Birkenheuer, K. Djemame, and O. Kao. "Risk Management Evaluation – Deliverable 1.2". Tech. Rep., October 2006.

[Mass 03] M. L. Massie, B. N. Chun, and D. E. Culler. "The Ganglia Distributed Monitoring System: Design, Implementation And Experience". Tech. Rep., University of Californi, Berkeley, Feb 2003.

[Maui 08] "Maui Scheduler Administrator Guide". http://www.clusterresources.com/products/maui/docs/mauiadmin.shtml, 2008.

[McCo 01] B. S. McConnell. *Beyond Contact: A Guide to SETI and Communicating with Alien Civilizations*. O'Reilly Media, March 2001.

[MOF 04] "MOF Risk Management Discipline for Operations". available at: http://www.microsoft.com/technet/solutionaccelerators/cits/mo/mof/mofrisk.mspx, 2004.

[Mold 06] J.-F. Molderez, C. Ponsard, I. Gourlay, K. Voss, J. Padgett, , K. Djemame, G. Birkenheuer, S. Mouton, and O. Kao. "Requirement Analysis – Deliverable 1.1". Tech. Rep., September 2006.

[Moll 06] E. Mollick. "Establishing Moore's Law". *IEEE Annals of the History of Computing*, Vol. 28, No. 3, pp. 62–75, 2006.

[Mont 03] J. Montagnat, V. Breton, and I. Magnin. "Using grid technologies to face medical image analysis challenges". In: *Biogrid'03, proceedings of the IEEE CCGrid03, Tokyo, Japan*, May 2003.

[NextGrid 08] "NextGRID – Project Homepage". http://www.nextgrid.org, 2008.

[Ngai 05] E. W. T. Ngai and F. K. T. Wat. "Fuzzy decision support system for risk analysis in e-commerce development". *Decis. Support Syst.*, Vol. 40, No. 2, pp. 235–255, 2005.

[Ober 05] B. Oberschmidt. "Risiko IT". In: *IM - Information Management & Consulting, 2/2005*, pp. 66–70, 2005.

[OGF 08] "Open Grid Forum (OGF)". http://www.ogf.org, 2008.

[OMah 05] D. O'Mahony. "IT-Risk-Management - Nicht Nur eine Aufgabe der IT-Abteilung". In: *IM - Information Management & Consulting, 2/2005*, pp. 61–65, 2005.

[OntoGrid 08] "Paving the way for Knowledgeable Grid Services and Systems (OntoGrid) – Project Homepage". http://www.ontogrid.net/, 2008.

[OpenCCS 08] "Cluster Computing Center (OpenCCS) – Project Homepage". http://www.openccs.eu, 2008.

[OpenPBS 08] "OpenPBS Homepage". http://www.openpbs.org, 2008.

[Ouel 05] D. Ouelhadj, J. Garibaldi, J. MacLaren, R. Sakellariou, and K. Krishnakumar. "A Multi-agent Infrastructure and a Service Level Agreement Negotiation Protocol for Robust Scheduling in Grid Computing´". In: *Advances in Grid Computing - EGC 2005, European Grid Conference, Revised Selected Papers*, pp. 651–660, Springer, Amsterdam, The Netherlands, Feb 2005.

[Panw 88] S. S. Panwar, D. Towsley, and J. K. Wolf. "Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service". *J. ACM*, Vol. 35, No. 4, pp. 832–844, 1988.

[ParWorkl 08] "Parallel Workload Archive – Homepage". http://www.cs.huji.ac.il/labs/parallel/workload/, 2008.

[PDSI 08] "Petascale Data Strorage Institute – Carnegie Mellon University". http://pdsi.nersc.gov/, 2008.

[Pich 08] A. Pichot, P. Wieder, O. Wäldrich, and W. Ziegler. "Dynamic SLA Negotiation based on WS-Agreement". *Proceedings of the 4th International conference on Web Information Systems and Technologies (WEBIST 2008), Funchal, Portugal (to appear)*, May 2008.

[Pinh 07] E. Pinheiro, W.-D. Weber, and L. A. Barroso. "Failure trends in a large disk drive population". In: *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, pp. 2–2, USENIX Association, Berkeley, CA, USA, 2007.

[Rain 91] R. K. Rainer and C. A. Snyder. "Risk Analysis for Information Technology". *Journal of Management Information Systems*, Vol. 8, No. 1, pp. 129–147, 1991.

[Raju 06] N. Raju, Gottumukkala, Y. Liu, C. B. Leangsuksun, R. Nassar, and S. Scott. "Reliability Analysis in HPC clusters". *Proceedings of the High Availability and Performance Computing Workshop*, 2006.

[Rama 95] R. Ramanathan, Ed. *Introductory Econometrics with applications*. The Dryden Press, Harcourt Brace College Publishers, Orlando, third Ed., 1995.

[Saha 01] A. Sahai, A. Durante, and V. Machiraju. "Towards Automated SLA Management for Web Services". Tech. Rep. Research Report HPL-2001-310 (R.1), Jul 2001. on-line at: http://www.hpl.hp.com/techreports/2001/HPL-2001-310R1.pdf.

[Saha 03] A. Sahai, S. Graupner, V. Machiraju, and A. van Moorsel. "Specifying and Monitoring Guarantees in Commercial Grids through SLA". In: *CCGRID '03: Proceedings of the 3st International Symposium on Cluster Computing and the Grid*, p. 292, IEEE Computer Society, Washington, DC, USA, 2003.

[Savv 07] A. Savva. "JSDL SPMD Application Extension – Version 1.0". August 2007. GFD-R-P.115 – JSDL-WG – Open Grid Forum (OGF).

[Scha 97] R. R. Schaller. "Moore's law: past, present, and future". *IEEE Spectr.*, Vol. 34, No. 6, pp. 52–59, 1997.

[Schn 05] E. Schnepf. "Grid Computing Solutions for Scientific and Business Critical Computing". Presentation hold on the Gridcoord Industrial Workshop and 8th HLRS Metacomputing and Grid Workshop, 2005.

[Scho 02] J. M. Schopf. "A General Architecture for Scheduling on the Grid". Tech. Rep. ANL/MCS-P1000-1002, 2002.

[Schr 06] B. Schroeder and G. A. Gibson. "A large-scale study of failures in high-performance computing systems". *DSN '06: Proceedings of the International Conference on Dependable Systems and Networks*, pp. 249–258, 2006.

[Seti 08] "Seti@Home". http://setiathome.berkeley.edu/, 2008.

[SGE 08] "Sun Grid Engine (SGE) Homepage". http://gridengine.sunsource.net, 2008.

[Sher 04] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya. "Libra: a computational economy-based job scheduling system for clusters". *Software – Practice & Experience*, Vol. 34, No. 6, pp. 573–590, 2004.

[Sivi 96] D. S. Sivia. *Data Analysis: A Bayesian Tutorial (Oxford Science Publications)*. Oxford University Press, July 1996.

[Smar 92] L. Smarr and C. E. Catlett. "Metacomputing". *Commun. ACM*, Vol. 35, No. 6, pp. 44–52, 1992.

[Srin 02] S. Srinivasan, R. Kettimuthu, V. Subrarnani, and P. Sadayappan. "Characterization of Backfilling Strategies for Parallel Job Scheduling". *icppw*, Vol. 00, p. 514, 2002.

[Srin 06] L. Srinivasan and T. Banks. "Web Services Resource Lifetime 1.2 (WS-ResourceLifetime)". http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf, April 2006. Document-Identifier: wsrf-ws_resource_lifetime-1.2-spec-os.

[Stew 04] A. Steward. "On risk: perception and direction". *Computers and Security*, pp. 362–370, 2004.

[Stoc 03] H. Stockinger, F. Donno, E. Laure, S. Muzaffar, P. Kunszt, G. Andronico, and P. Millar. "Grid Data Management in Action: Experience in Running and Supporting Data Management Services in the EU DataGrid Project". 2003.

[Symantec 08] "IT Risk Management Report 2: Myths and Realities – Trends through December 2007". Tech. Rep. 2, Jan 2008.

[Talb 99] D. Talby and D. G. Feitelson. "Supporting Priorities and Improving Utilization of the IBM SP2 Scheduler Using Slack-Based Backfilling". In: *13th Intl. Parallel Processing Symp.*, pp. 513–517, 1999.

[Tali 02] D. Talia. "The Open Grid Services Architecture: Where the Grid Meets the Web". *IEEE Internet Computing*, Vol. 6, No. 6, pp. 67–71, 2002.

[Tsay 05] R. S. Tsay. *Analysis of financial time series*. John Wiley & Sons, New York, second Ed., 2005.

[Tuec 03] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. VAnderbilt. "Open Grid Services Infrastructure (OGSI)– Version 1.0". Jun 2003. GWD-R (draft-ggf-ogsi-gridservice-33) – OGSI WG – Open Grid Forum.

[Unicore 07] "UNICORE Forum e.V.". http://www.unicore.org, 2007.

[Voss 06] K. Voss. "Risk Aware Migrations For Prepossessing SLAs". *International Conference on Networking and Services (ICNS)*, Vol. 0, p. 68, 2006.

[Voss 07a] K. Voss. "Comparing Fault Tolerance Mechanisms for Self-Organizing Resource Management in Grids". In: *SKG '07: Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, pp. 50–55, IEEE Computer Society, Washington, DC, USA, 2007.

[Voss 07b] K. Voss. "Enhance Self-managing Grids by Risk Management". In: *ICNS '07: Proceedings of the Third International Conference on Networking and Services*, p. 27, IEEE Computer Society, Athens, Greece, 2007.

[Voss 07c] K. Voss, D. Battré, O. Kao, and K. Djemame. "Gaining Users' Trust by Publishing Failure Probabilities". In: *Proceedings of the First International Workshop on Security, Trust and Privacy in Grid Systems (Grid-STP'2007)*, Nice, France, Sep 2007.

[Voss 07d] K. Voss, K. Djemame, I. Gourlay, and J. Padgett. "AssessGrid, Economic Issues Underlying Risk Awareness in Grids.". In: J. Altmann and D. Veit, Eds., *GECON*, pp. 170–175, Springer, 2007.

[Voss 07e] K. Voss, I. Gourlay, J. Padgett, D. Battré, , and M. Quijada. "AssessGrid 1st Prototype – Deliverable 2.1". Tech. Rep., Oct 2007.

[Voss 08a] K. Voss. "Recursive Evaluation of Fault Tolerance Mechanisms for SLA Management". *International Conference on Networking and Services (ICNS) 2008*, Vol. 0, pp. 223–229, 2008.

[Voss 08b] K. Voss and C. Carlsson. "AssessGrid Risk Assessment and Consultant Service – Deliverable 3.1". Tech. Rep., Apr 2008.

[Vyss 65] V. Vyssotsky, M. Hill, F. J. Corbato, and R. M. Graham. "Structure of the Multics Supervisor". *AFIPS Conference Proceedings*, Vol. 27, No. 1, pp. 203–212, 1965.

[Wald 08] O. Wäldrich. "WS-Agreement Framework (WSAG4J)". http://packcs-e0.scai.fhg.de/mss-project/wsag4j/, 2008.

[Welc 03a] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. "Security for Grid Services". In: *High performance distributed computing*, pp. 48–57, IEEE Computer Society Press, 2003.

[Welc 03b] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. "Security for Grid Services". *HPDC '03: Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, p. 48, 2003.

[Whit 95] D. White. "Application of systems thinking to risk management: a review of the literature". *Management Decision*, pp. 34–45, 1995.

[Witt 05] H. M. Wittmann. "IBM Grid Computing & Virtualization - Trends and Directions". Presentation hold on the Gridcoord Industrial Workshop and 8th HLRS Metacomputing and Grid Workshop, 2005.

[Wrze 05] G. Wrzesinska, R. van Nieuwpoort, J. Maassen, and H. E. Bal. "Fault-Tolerance, Malleability and Migration for Divide-and-Conquer Applications on the Grid.". In: *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, p. 13.1, 2005.

[Yeo 05] C. S. Yeo and R. Buyya. "Service Level Agreement based Allocation of Cluster Resources: Handling Penalty to Enhance Utility". *IEEE International Conference on Cluster Computing*, Vol. 0, pp. 1–10, 2005.

[Zwic 67] F. Zwicky and A. Wilson. "New Methods of Thought and Procedure". In: *Contributions to the Symposium on Methodologies*, Springer, Berlin, 1967.

[Zwic 69] F. Zwicky. "Discovery, Invention, Research Through the Morphological Approach". In: *Contributions to the Symposium on Methodologies*, McMillan, New York, 1969.

[Zwic 98] F. Zwicky. "Morphology and Policy Analysis". In: *Proceedings of the 16th EURO Conference on Operational Analysis*, Brussels, 1998.